

Б.М. МҰстафина, М.А. Ахметова

# MS DOS – ТЫҚ ЖҮЙЕЛІК ФУНКЦИЯЛАРЫ

Оқу құралы

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ  
МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы  
Қазақ ұлттық техникалық университеті

Б.М. Мұстафина, М.А. Ахметова

## MS DOS – ТЫҚ ЖҮЙЕЛІК ФУНКЦИЯЛАРЫ

Оқу құралы

Республикалық оқу-әдістемелік бірлестігі оқу құралы ретінде  
ұсынған

АЛМАТЫ 2002

ЖОК 681.3.06:519.682.8 (075)

ББК 32.973-01я7

М78

М78 **Мұстафина Б.М., Ахметова М.А.**

MS DOS-ты” жүйелік функциялары. Оқу құралы. Алматы: Қаз ҰТУ, 2003. 125б.

ISBN 9965-585-78-4

Оқу құралы мамандарды” білім дәрежесіне қойылатын талаптарға, мемлекеттік стандарттарға, дәрістерді оқуды” педагогика-психологиялық негіздеріне сәйкес жасалынған. Оқу құралында MS DOS операциялық жүйені тиімді қолдану әдістері, ерекшеліктері және мүмкіндіктері туралы түсінік беріледі. Материалда MS DOS-ты” жиі қолданылатын функциялары сипатталған және осы жүйелік құралдарда қолданбалы программаларды қолдану мысалдары келтірілген.

Материал Жүйелік программалау пәніні” негізгі бөлімдерін қамтиды. Бұл е”бекті анықтама құралы ретінде де қолдануға болады.

Оқу құралы 3701 ("Есептеу машиналар, жүйелік және тораптар") және 3706 ("Ақпаратты қорғау және қауіпсіздендіру") мамандықтарыны” студенттеріне арналған.

Сурет – 13. Кесте – 13. Әдебиеттер тізімі – 8 атау

ББК 32.973-01я7

Пікір жазғандар: *Қуандықов А.А.* ҚР ҰБ АТД  
басқармасыны” орынбасары,  
техника ғыл. канд.,  
*Өскенбаева Р.К.* Қаз ҰТУ  
"Ақпараттандыру" кафедрасыны”  
доценті, тех.ғыл. кандидаты.

Қазақстан Республикасы Білім және ғылым министрлігіні”  
басылым жоспары бойынша 2003 басылды

М  $\frac{4301020000}{00(05) - 03}$

ISBN 9965-585-78-4

© Қаз ҰТУ, 2003

## **КІРІСПЕ**

Intel процессор негізінде жасалған компьютерлерде е” қарапайым және ке” таралғаны - DOS (Дискілік операциялық жүйе) операциялық жүйесі. Оны өндіруші фирмалар қатарына мыналар жатады: Microsoft (MS DOS), IBM (PS DOS), Novell (Novell DOS), Calderd (Open DOS).

Бұл жүйе Microsoft Windows 95 немесе одан да жоғары жүйесіні” бөлігі ретінде беріледі. DOS программаларды” жұмыс істеуіне толық еркіндік береді, яғни сыртқы құрылғылармен және жадымен шексіз жұмыс істеуге және сол құрылғылар арқылы процессормен басқаруға және жадыны бөлуге мүмкіндік береді. Сол себептен DOS компьютер құрылғыларымен жақын танысу үшін е” жақсы мүмкіндік беретін жүйеге жатады. DOS жүйесі ассемблер тілінде жазылған программаларды пайдалануға мүмкіндіктер береді. Басқа жетілген операциялық жүйелер мен жоғарғы тілдегі компиляторлар бұл мүмкіндікті жасырып тұрады.

Операциялық жүйені” е” ма”ызды сы”арларына операциялық жүйені” функциялары деп аталатын, қызмет ететін жүйелік программалар жатады. Олар компьютерды” жабдықтарын басқару және есептеу процесін ұйымдастыру ретінде қарапайым әрекеттерді орындайды. Операциялық жүйені” қызметтері көмегімен файлдық жүйені” жұмысы, пернетақтадан ақпаратты енгізу, бейнеконтроллерге қызмет ету, жадыны динамикалық ерекшелеу және босату, туынды программаларды динамикалық іске қосу, жадыда резидентті программаларды құру, драйверлерді және үзу ө”деуіштерін құру сияқты жұмыстар орындалады.

Microsoft Windows-қа арналған қолданбаларын жасаумен айналысатын программалаушыларға да MS DOS-ты” программалық интерфейсі және жұмыс принциптері ретінде жақсы білім қажет болады. Оны” себебі Microsoft Windows және MS DOS операциялық жүйелері бір-бірімен тығыз әрекеттесіп жұмыс істейді.

## 1 DOS-ты” ҚҰРЫЛЫМЫ МЕН ЖҰМЫС ПРИНЦИПТЕРІ

### 1.1. Операциялық жүйені” негізгі сы”арлары

MS DOS-ты” негізгі сы”арларына **io.sys**, **msdos.sys** және **command.com** файлдары жатады.

**Io.sys** файлы құрамында **BIOS** (Basic Input/Output System) енгізу/шығару негізгі жүйені” ке”ейтуі бар, ол BIOS пен операциялық жүйені” арасындағы интерфейсі болады. Бұл ке”ейту операциялық жүйені” компьютерді” аппаратурасымен өрекеттесуі үшін қолданылады.

Программаларды” тәуелділігін компьютерді” аппараттық қамтаманы” ерекшеліктерінен максималды азайту үшін, программалардан енгізу/шығару құрылғыларына қатынауды” көпсатылы сұлбасы қолданылады. Бұл сұлба бойынша программа аппаратура порттарына ешуақытта да тікелей қатынаспайды. Оны” орнына программа керек нөмірмен программалық үзуді шақырады да, процессорды” оған сәйкес регистрлерін жүктеп, керек функцияны орындайды.

Барлық программалық үзулерді екі топқа бөлуге болады.

Бірінші топқа BIOS енгізу/шығару негізгі жүйені” үзулері жатады. Бұл үзу ө”деуіштері тұрақты еске сақтайтын құрылғысыны” микросұлбасында жазылған. BIOS-ты” үзу ө”деуіштері компьютерді” стандартты құрылғыларыны” енгізу/шығару порттарымен жұмыс істейді және оған сәйкес аппараттық үзулерді қамтамасыз етеді.

Екінші топқа DOS үзулері жатады. Бұл үзу ө”деуіштері **io.sys** және **msdos.sys** файлдарында болады.

MS DOS-ты” енгізу/шығару негізгі жүйені” ке”ейтуі **Io.sys** файлда орналасқан, MS DOS пен BIOS-ты” арасында аралық қабат болып табылады.

**Msdos.sys** файлы үзулерді ө”дейтін программаларды” жиынын қосады, мысалы **INT 21h** үзуін. Бұл операциялық жүйені” ядросы.

**Command.com** файлы құрамында компьютерді” пайдаланушымен сұхбатын ұйымдастыру үшін командалық процессор болады. Ол енгізілген командаларды талдайды және оларды” орындалуын ұйымдастырады. **Command.com** файл құрамында ішкі командаларды” ө”деуіштері бар.

Командалық процессор үш бөлімнен (резидентті, инициализациялайтын және транзитті) тұрады. Резидентті бөлім

жадыға бірінші жүктелінеді. Ол INT 22h, INT 23h, INT 24h үзулерді ө”дейді, транзитті бөліміні” жүктеуін басқарады. Транзитті бөлім қателерді ө”дейді және қателер табылғанда қандай әрекеттер керек екені туралы пайдаланушыдан сұрайды.

Инициализациялайтын бөлім тек қана операциялық жүйені жүктеу процесінде пайдаланылады. Бұл бөлім іске қосылатын программаларды жадыға жүктеуді” бастапқы адресін анықтайды және autoexec.bat файл орындалуын инициализациялайды.

Командалық процессорды” транзитті бөлімі жадыда үлкен адресстер бойынша орналасады. Бұл бөлікте MS DOS-ты” ішкі командаларыны” ө”деуіштері және дестелік файлдарды” (атыны” .bat ке”еюі бар) интерпретаторы болады. Жиі қолданылатын MS DOS командалар тізбектерін автоматты түрде орындау үшін дестелік деп аталатын файлды құруға болады. Осындай файлды” әр қатарында MS DOS-ты” ішкі немесе сыртқы команданы” шақыруы немесе арнайы операторлары бар болады.

Транзитті бөлім жүйелік шақыруды шығарады, операторды” командасын пернетақтадан немесе дестелік файлдан енгізуін күтеді және оны” орындауын ұйымдастырады.

MS DOS-ты орнатқан со” түпкі каталогта жоғарыда аталып кеткен файлдардан басқа **config.sys** және **autoexec.bat** аттары бар екі мәтіндік файл құрылады.

Config.sys файлы MS DOS-ты” конфигурациясын анықтау және драйверлер мен резидентті программаларды жүктеу үшін тағайындалған.

Autoexec.bat файлы MS DOS-ты жүктеген со” бірден іске қосылады. Бұл файл құрамында программаларды орындау ортасыны” анықтау командалары, сонымен бірге инициализациялайтын және резидентті программаларды” шақыруы бар.

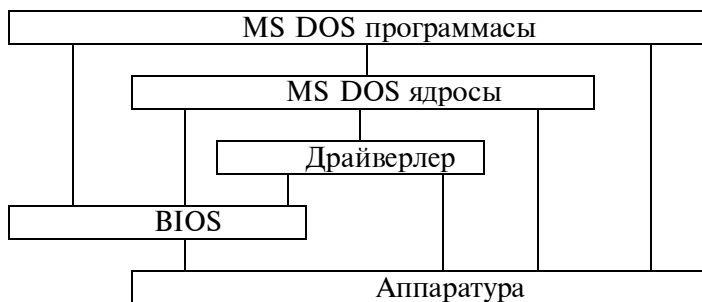
Қандай да бір аппаратураны қамтамасыз ететін программаларды - драйвер деп атайды. Көбінесе драйверлер .sys ке”ейтуі бар файлдарда орналасады, бірақ та MS DOS-та құрастырылған драйверлер де бар. Драйвер MS DOS операциялық жүйеге config.sys файлда орналасқан **device** операторы көмегімен қосылады. Жа”а құрылғыны қосқанда қолданбалы программаларға және операциялық жүйеге өзгертулерді енгізу қажет емес, өйткені олар құрылғылармен драйвер арқылы әрекеттеседі.

MS DOS-ты” жай программаларыны” жұмыстары біткеннен со”, программа жедел жадыдан бірден жойылады. Одан өзгеше резидентті программалар жадыда қалады және оларды алдын ала

берілген пернелер комбинациясы көмегімен, басқа программаны” талабы бойынша немесе аппараттық үзуі пайда болу нәтижесі ретінде жандандыруға болады.

## 1.2 MS DOS-ты” құрылымы мен жұмысыны” жалпы сұлбасы

Жүйелік программалық және аппараттық қамтамаларымен дұрыс жұмыс істеу үшін MS DOS операциялық жүйені” және компьютерді” аппаратурасыны” қолданбалы программамен әрекеттесу механизмін түсіну керек (1.1 сурет).



1.1- сурет. Программаны” MS DOS, BIOS және компьютерды” аппаратурасымен әрекеттесу сұлбасы

MS DOS программасы MS DOS ядросына, BIOS енгізу/шығару негізгі жүйесіне, сонымен бірге аппаратураны” енгізу/шығару порттарына тікелей қатынаса алады. Бірақ та программа MS DOS драйверлерін тікелей шақыра алмайды.

MS DOS ядросы драйверлер мен BIOS үзулерді шақыра алады және аппаратурамен жұмыс істей алады. Өз кезегінде драйверлер енгізу/шығару порттарымен жұмыс істейді де BIOS программалық үзулерді шақыра алады.

MS DOS ядросы бірнеше жүйеге (файлдық, жадыны басқару, программаларды басқару, құрылғылар драйверлері-мен байланыс, қателерді ө”деу, уақыт қызметі, енгізу/шығару консолі) бөлінеді.

Бұл жүйелер аппаратурамен тікелей немесе BIOS үзулері, драйверлер арқылы қатынасады.

Қолданбалы программалық қамтама MS DOS жүйелерін шақыра алады, BIOS-пен немесе аппаратурамен тікелей жұмыс істей алады.

### **1.2.1 Файлдық жүйе**

Операциялық жүйе жүйелік дискіде (io.sys, msdos.sys, command.com драйверлері, сыртқы командалар және т.б.) файлдар түрінде жазылған. Барлық программалар және оларға арналған деректер де файл түрінде дискілерде сақталады.

Файлдық жүйеге қатынасу үшін қолданбалы программа MS DOS-ты” программалық үзулеріні” арнайы функцияларын қолдануға міндетті. Осындай функциялар барлық файлдық операцияларды: - файлдарды және каталогтарды құру немесе жою, буферленген және буферленбеген жазу немесе оқу, файлдық жүйе туралы анықтамалық ақпаратты алу және т.с.с. орындайды.

Файлдық жүйе дискілермен драйверлер арқылы жұмыс жасайды. Өз кезегінде драйверлер BIOS қызметтерін пайдаланады.

BIOS де”гейінде секторларды оқу/жазу, бөліп белгілеу және тағы басқа осы сияқты дискімен қарапайым операциялар орындалады. Бұл төменгі де”гейді қолданбалы программаларды да пайдалана алады, бірақ та көбінесе программа каталогтарды және файлдарды орналастыру кестелерін (File Allocation Table - FAT) қамтамасыз ету ретінде барлық керек әрекеттерді орындайтын MS DOS үзулеріні” функцияларын пайдаланады.

Рұқсатсыз қатынаудан немесе көшіруден қорғау программалары BIOS үзулерін шақырады және төменірек де”гейді” құралдарына қатынасуға немесе енгізу/шығару порттары арқылы тіпті диск контроллермен жұмыс істеуге мәжбүр болады.

### **1.2.2 Жадыны басқару жүйесі**

MS DOS бұл жүйені іске қосылатын программаларға арнап, жадыны бөліп үлестіру үшін қолданады.

MS DOS MCB (Memory Control Block) блоктар көмегімен жадыны басқарады. Жады үзінділерге бөлінеді. Әрбір үзіндіні” алдында жады блогыны” сипаттамалары жазылған MCB орналасады. Әрбір жа”адан іске қосылатын программаға арнап MS DOS белгілі бір сан арқылы жады үзінділерін құрады. MS DOS жадыны басату немесе қосымша жадыны алу ретінде сұратуды орындау кезінде MCB блоктарыны” мазмұнын тексереді және соған сәйкес ол мазмұнды өзгертеді.

Барлық MCB блоктары біріні” артынан бірі орналасады. Бірінші блокты” адресі CVT байланыс векторлық кестесінде

сақталады. Бұл кестеге құжатталмаған функциялар арқылы қатынауға болады.

Қолданбалы программа өзі үшін жадыдан қосымша үзінділерді сұрауына болады. Осы сұратуды орындау үшін программа INT 21h үзуіні” функцияларын қолданып, жадыны басқару жүйесіне қатынасып сұрау салады.

### **1.2.3 Программаларды басқару жүйесі**

MS DOS операциялық жүйе программаны іске қосқанда бірнеше әрекеттер орындайды. Алдымен MS DOS қолданбалы программаға арнап жады блоктарын дайындау үшін, жадыны басқару жүйесімен қатынасады. Файлдық жүйе көмегімен құрамында программасы бар файл жадыға жүктеледі, сосын программа (бұл тек .exe типті файлдарына ғана қатысты) нақты физикалық адреске бапталынады. Тек содан кейін ғана программаға басқару беріледі.

MS DOS-та орындалатын файлды” екі түрі бар. Оларды” аттарыны” .com және .exe ке”ейтулері бар. Бұл программаларды іске қосу әдістері ерекше болып келеді. Программаларды басқару жүйесі оларды автоматты түрде тауып, екеуін екі түрлі жадыға жүктейді.

Программаларды басқару жүйесіндегі есептелетін басқа есептер қатарына - программадан программаны” іске қосылуы және оверлейлерді” (жадыға қалқалаумен жүктелінетін программаны” қосымша модульдері) жадыға жүктелуі сияқты қызметтер жатады. Егер бір мезгілде үлкен программалық комплексті” модульдері керек болмаса, онда комплексті бірнеше бөлікке бөлулері”ізге болады. Ол бірнеше бөлек программалар немесе бірнеше оверлейлі модульдер болуы мүмкін. Бұларды” әрқайсыны” кемшілігі мен артықшылықтары бар. Оларды” екеуі де жадыны аз қолданады.

Программаларды басқару жүйесіні” со”ғы функциясы - резидентті программамен жұмыс істеу. Егер сіз программа жұмысы біткен со” оны резидентті түрінде қалдырғы”ыз келсе, онда сіз жай программа аяқталғанда DOS-ты” соған байланысты функциясы арқылы программаларды басқару жүйесіне қатынаса аласыз.

#### **1.2.4. Құрылғылар драйверлерімен байланыс жүйесі**

Бұл жүйе қолданбалы программалардан жасырылған. Программалар өз билігімен шалғай құрылғылар драйверін қолдана алмайды. Программа DOS үзулерін шақырады, ал ол қажет болғанда драйверлермен қатынаса алады.

MS DOS-ты” кейбір құжатталмаған үзулерді қолданып өз билігімен драйверды шақыруына болады, бірақ онда операциялық жүйені” басқа нұсқаларыны” (версия) арасындағы бірлестігінен айырылуы мүмкін. Енгізу/ шығаруды” жағдайын және драйверді” жағдайын басқару үшін INT 21h үзуіні” арнайы 44h функциясы қолданылады. Бұл функция қолданбалы программа мен драйверді” арасындағы басқарудағы ақпаратты ауыстыруға арналған.

#### **1.2.5. Қателерді ө”деу жүйесі**

MS DOS үзулерін шақыру кезінде пайда болатын қателерді табу үшін, ереже бойынша процессорды” тасымал белгісі (жалаушасы) қолданылады. Егер MS DOS үзуімен қатынарудан кейін тасымал жалаушасы бірге те” болса, онда қате болды.

Қатені талдау және қандай да бір әрекетті жасау үшін, қате туралы анықтау ақпараттын беретін және соған сәйкес нұсқау (процессорды” регистрлерінде жазылған кодалар түрінде) ұсынатын MS DOS-ты” қажетті функциясын шақыруға болады.

#### **1.2.6. Уақыт қызметі**

Кез келген компьютер жүйелік сағатпен жабдықталған. Бұл сағат аккумулятордан қоректенеді. Компьютерді өшіргенде сағатты” мазмұны тазаланбайды.

Қосымша ретінде уақыт аралықтарын қалыптастыру үшін үзулерді үнемі ө”діретін таймер пайдаланылады. Қандай болса да функцияларды” реттік қызметін орындау үшін программа таймерді” үзулерін қолдана алады. MS DOS операциялық жүйесі уақытты есептейді және сағатты” ағымдағы көрсеткіші мен ағымдағы күнін сақтайды.

Программа INT 21h-ты” керек функциясы арқылы MS DOS-қа қатынасып сағатты сұрай алады немесе сағатты” жа”а мәнін орната алады. MS DOS операциялық жүйе құрамында CLOCKS\$

құрылғы драйвері бар. Қолданбалы программа сағат көрсеткішін оқу немесе сағатты орнату үшін бұл құрылғыға қатынаса алады.

### 1.2.7. Оператор консолімен жұмыс

Оператор консолі екі құрылғыдан (пернетақта және дисплей) тұрады. Бұл екі құрылғыны” CON атты бір консоль драйвері болады. Операциялық жүйе CON құрылғысына символдарды” енгізуін және шығаруын INT 21h үзуіні” функциялары көмегімен орындайды. Пернетақтамен және бейнеконтроллермен жұмыс істеу үшін, бұл драйвер BIOS үзулерін шақырады.

### 1.2.8. DOS-ты” басқа жүйелері

MS DOS-та тізбекті және параллель порттарымен жұмыс істеу үшін драйверлері бар. Олар символдық режимінде деректерді” алмасуын және оларға қосылған құрылғыларды” (әдетте бұл принтер, маус немесе факс-модем) басқаруын қамтамасыз етеді.

Одан басқа MS DOS құрамына мәтінді фондық режимде баспаға шығару үшін **print.com** программасы кіреді. Бұл іс жүргізудегі жалғыз "мультиесепті" утилита.

Электронды тегеріш немесе кэш-жадыны ерекше айтып өту керек. Электронды дискіні” драйвері **ramdrive.sys** деп аталады. Бұл драйвер ке”ейтілген немесе қосымша жадыда жедел жалған диск ұйымдастырады. Электронды дискіні негізгі жады да ұйымдастыруға болады, бірақ негізгі жады әрқашан аз болады.

MS DOS операциялық жүйесіні” құрамында himem.sys және emm386.exe драйверлерімен жүзеге асырылатын қосымша және ке”ейтілген жадыны басқару жүйесі бар. Бұл драйвер қосымша және ке”ейтілген жадыны қолдануға мүмкіндік береді.

Егер **himem.sys** драйвері қосылған болса, онда **smartdrv.sys** драйверіні” көмегімен қатты диск үшін бүркеме-жадыны (кэш-жад) құруға болады. Әсіресе бүркеме-жадыны қолдану сол бір ғана ақпарат мерзімді қажет болғанда деректер базасымен жұмыс істеу үшін тиімді. Келесіде ақпарат қажет болғанда дискіге қатынас жасамай-ақ, оны алуға болады, ол өзі жеткізіледі.

Басқа MS DOS-пен қолдау жасалмайтын құрылғыларды фирма-өндірушілер арнайы программалармен, драйверлермен қамтамасыз етіп тұрады және оларды операциялық жүйеге қосып отыру керек.

## 2. ҚОЛДАНБАЛЫ ПРОГРАММАДА ЖҮЙЕЛІК ҚҰРАЛДАРДЫ ПАЙДАЛАНУ

DOS пен BIOS-ты” қызметтеріне қатынас құру ұзу арқылы (INT командасы) жүзеге асырылады.

Үзуді” түрлері деп аталатын нөмірлері болады. Үзуді” түрлері бойынша оны ө”дейтін программаны” адресі сақталатын, жадыны” төрт байттық аймағы, яғни үзуді” векторы есептеледі.

Ұзу векторларыны” кестесі жедел жадыны” 0000h бастап 03FFh адресіне дейін орналасады. Әр векторды” екі үлкен байтында ҮӨП (Ұзу Ө”дейтін Программа) сегментті” адресі, ал екі кіші байтында - сегментті” ішіндегі ҮӨП адресі (ығысу адресі) орналасқан.

N нөмірлі вектор (N\*4)-тен (N\*4+3) байтқа дейінгі жадыны иемденеді (мұндағы, N-үзуді” түрі). Жадыны” векторларға бөлінген аймағында 256 вектор орналасады. Үзуді” орындалу сұратуына қойылатын сауалдар әр түрлі болып келеді. Атап айтқанда сыртқы құрылғылардан немесе жүйені” басқа сы”арларынан аппараттық үзулер және INT командасы арқылы шақырылатын, DOS-ты” және BIOS-ты” қызметтеріне қолданылатын программалық үзулер деп бөлінеді.

Кез келген ұзу пайда болғанда процессор стекте жалаушалар регистріні”, CS және IP регистрлеріні” мәндерін сақтап, ұзу ө”дейтін программаны” адресін CS және IP-ге жазады да оған басқару береді. Ұзу ө”дейтін программа IRET үзуден қайта оралу командасымен аяқталады. IRET командасы IP, CS және жалаушалар регистрлерін стектен оқыған сөздермен толтырады. Сонымен басқару негізгі программаға беріледі.

Пайдаланушыға келесі нөмірлері бар векторларды қолдануға болады:

00h- нөлге бөлу;

01h- командаларды қадам бойынша орындау (егер TF=1 болса);

02h- қалқалы емес үзілім (процессорды” NMI шығысы);

03h- сан параметрі жоқ INT командасы;

04h- INTO - толып кету бойынша үзілім (программада INTO командасы болса);

05h - Print Screen пернені басқанда үзілім қоздырылады;

08h - таймер (аппараттық үзілім);

09h - пернетақта (аппараттық үзілім);

- 0Ah - стандартты емес құрылғыларды қосу үшін қосалқыланған (аппараттық үзілім);
  - 0Bh - COM2 екінші тізбекті порт (аппараттық үзілім);
  - 0Ch - COM1 бірінші тізбекті порт (аппараттық үзілім);
  - 0Dh - LPT2 екінші параллель порт (аппараттық үзілім);
  - 0Eh - иілгіш диск (аппараттық үзілім);
  - 0Fh - параллель порт (LPT1 принтер) (аппараттық үзілім);
  - 10h - BIOS бейнедрайвері;
  - 11h - компьютер конфигурациясы;
  - 12h - жедел жадыны” өлшемі;
  - 13h - дискіні” BIOS драйвері;
  - 14h - тізбекті портыны” драйвері;
  - 15h - ке”ейтілген қызмет;
  - 16h - пернетақтаны” BIOS драйвері;
  - 17h - принтерді” BIOS драйвері;
  - 19h - BIOS-ты” бастапқы жүктеуіші;
  - 1Ah - BIOS-ты” календарь-сағаты;
  - 1Bh - <Ctrl>/<Break> бойынша үзу ө”деуіші;
  - 1Ch - таймерді” аппараттық үзу ө”деуішпен бір секундта
- 18,2 рет шақыратын программалық үзілімі;
- 1Dh - BIOS, бейнепараметрлер кестесіні” адресі;
  - 1Eh - BIOS, дискетті” параметрлер кестесіні” адресі;
  - 1Fh - BIOS, 4...6 графикалық режимдерді” қаріптер кестесіні” екінші жартысыны” адресі;
  - 20h - программа жұмысыны” аяқталуы;
  - 21h - DOS қызметтері (функциялары);
  - 22h - DOS процесін аяқтау кезінде көшу адресі;
  - 23h - <Ctrl>/C бойынша үзу ө”деуіші;
  - 24h - критикалық қате бойынша үзу ө”деуіші;
  - 25h - дискіден абсолютті оқу;
  - 26h - дискіге абсолютті жазу;
  - 27h - программа жұмысын аяқтап, резидентті түрде оны жадыда қалдыру;
  - 28h - пернетақтадан командаларын енгізіп, резидентті программаларды жандандыру үшін DOS программасы;
  - 2Eh - DOS командасын орындау;
  - 2Fh - DOS-ты” мультиплексті үзілімі;
  - 33h - Microsoft фирманы” маус драйвері;
  - 43h - BIOS, графикалық режимдерді” қаріптер кестесіні” адресі;

60h...66h - пайдаланушыны” үзулері;  
67h - LIM EMS қосымша жадыны” драйвері;  
68h...6Fh - бос векторлар;  
70h - нақты уақытты” КМОП-сағаты (аппараттық үзілім);  
71h - стандартты емес сыртқы құрылғыларды қамтамасыз ету үшін BIOS-ты” INT 0Ah үзілімі;  
72h...73h - қосалқыланған;  
74h - маус (PS/2-ға арналған аппараттық үзілім);  
76h - қатты диск (аппараттық үзілім);  
77h - қосалқыланған;  
78h...7Fh - бос векторлар;  
F1h...FFh - пайдаланылмайды.

Үзуді” векторларын шартты түрде төмендегідей топтарға бөлуге болады :

- аппараттық үзулерді” векторлары (08h...0fh, 70h...77h);
- BIOS- ты” драйверлері (10h, 13h, 16h т.т);
- DOS-ты” программалары (21h, 22h, 23h т.т)
- DOS пен BIOS-ты” жүйелік кестелеріні” адрестері (1Dh, 1Eh, 43h..)

Адрестері үзулерді” векторларында сақталатын жүйелік программалар көп жағдайда жүйелік қызмет атқаратын программаларды” үлкен топтарына шығаратын диспетчерлер болып табылады.

Қолданбалы программадан жүйелік қызметтерге қатынасу біржақты іске асады. АН регистріне қызметті” нөмірлері жіберіледі (үзуді” түрімен шатастыруға болмайды!), ал басқа регистрлерге оны” жүйелік программасыны” орындалуына керекті берілген мәндер жіберіледі. Осыдан кейін үзуді” түрін көрсететін сандық аргументі бар INT командасы орындалады. DOS пен BIOS-ты” көптеген қызметтері CF тасымал жалауына аяқталу кодын қайтарады. Егер қызмет сәтті орындалса, CF=0 болады да , қате болса CF=1 болады. CF=1 болған жағдайда регистрлерді” біреуіне ( көбінесе AX регистріне ) қатені” кодасы қайтарылады.

Сонымен жүйелік құралдарға қатынасуды” тәртібі былайша бейнеленеді:

```
mov     АН, fun ; fun-қызметті” нөмірі
; (AL, BX, ES т.б) регистрлерді осы қызметті
; орындауға керекті параметрлермен толтыру
...
```

```
int      21h          ; MS-DOS-қа көшу
jc       error
; программаны” жалғасы
Error:
cmp      AX, 1        ; аяқталу кодын зерттеу
je       err1
cmp      AX, 2
je       err2
. . .
```

Осындай тәртіппен BIOS-ты” да қызметтері шақырылады.

## **2.1. BIOS үзулеріне шолу**

Барлық үзулер нөмірлері бірнеше топқа бөлініп, шалғай құрылғылардан аппараттық үзулеріне, BIOS үзулеріне, DOS-қа қосалқыланған, пайдаланушы программаларына берілген. Оларды” е” ма”ызды BIOS үзулерін қарастырайық.

### **2.1.1. Компьютерге қосылған жабдықты” жинағын анықтау**

Жабдық туралы ақпаратты талдау арқылы программа компьютерге иілгіш магниттік дискідегі жинақтауыш қосылған ба, жоқ па, қосылған жағдайда, оны” саны қанша екенін анықтай алады.

Сонымен қатар бейнеконтроллерді” алғашқы режімін анықтауға болады, арифметикалық сопроцессор бар ма, жүйеде қанша параллель және тізбектелген порттар орнатылған, ойын адаптері іске қосылған ба, осы сияқты сұрақтарды анықтап білуге болады.

Әдетте программаны” өзі аппаратурамен жұмыс істемейді, ол операциялық жүйе қызметін пайдаланады. Операциялық жүйе көмегімен стандартты аппаратураға сұраныс жасағанда, егер ол құрылғы болмаса, онда программа қате белгісін алады.

Компьютерге қосылған жабдықты” жинағын анықтау үшін INT 11h қолдануға болады. Жабдық жайында ақпарат AX регистрінде қайтарылады. Әрбір AX регистрді” разряды сәйкес құрылғы бойынша жауапты болады.

| Разряд | Мәні | Сипаттамасы                                  |
|--------|------|----------------------------------------------|
| 0      | 1    | Иілгіш диск орнатылған                       |
| 1      | 1    | Арифметикалық сопроцессор орнатылған         |
| 2,3    |      | Жедел жадыны” көлемі:                        |
| 4-5    |      | Бастапқы бейне режимі                        |
|        | 01   | 1 режим (40*25)                              |
|        | 10   | 3 режим (80*25)                              |
|        | 11   | 7 режим (80*25)                              |
| 6-7    |      | Дискжетектер саны (0 разряд 1-ге те” болса): |
|        | 00   | 1                                            |
|        | 01   | 2                                            |
|        | 10   | 3                                            |
|        | 11   | 4                                            |
| 8      |      | Қосалқыланған                                |
| 9-11   |      | Орнатылған RS-232 порттар саны               |
| 12     | 1    | Ойын адаптер орнатылған                      |
| 13     |      | Қосалқыланған                                |
| 14-15  |      | Орнатылған принтер саны                      |

Бұл ақпаратты 0000:0410h адресі бойынша сөзінен алуға болады.

### 2.1.2. Жедел жады көлемін анықтау

Кейбір программаларды” жұмыс істеуі үшін оған қажетті жады көлемі керек. INT 12h үзуі AX регистріне көлемі 1 Кбайт болатын жадыдағы бар блоктар санын қайтарады. Программалар осы шаманы талдайды. Егер жады көлемі аз болса, экранға соған сәйкес хабар шығарып, жұмыс істеуден бас тартады немесе жұмыс істеу алгоритмін өзгертеді, мысалы дискідегі виртуалды жадыны ұйымдастырады не аралық нәтижелерді файлға жазып отырады. Егер компьютерде ке”ейтілген жады болса (ке”ейтілген жадыны” адреcтік ке”істігі 1Мбайттан көп болады), онда оны” көлеміні” килобайттық мөлшерін AX регистрі 8800h-те” INT 15h үзуін шақыру арқылы білуге болады.

### 2.1.3. Бейнеконтроллермен жұмыс

INT 10h үзуі бейнеконтроллерге қызмет ету операцияларын орындайды.

#### ***Бейнережимді та"дау (00h)***

Бұл функция әдетте бейнеконтроллермен жұмысты" басында е" бірінші болып шақырылады немесе бейнеконтроллерді" ағымдағы режимін өзгерту үшін шақырылады.

Бейнеконтроллер мәтіндік немесе графикалық режимде жұмыс істейді. Мәтіндік не графикалық режимдерді" көптеген түрлері кездеседі. Оларды" бір-бірінен айырмашылығы бейнелеуді" пікселдік мөлшері (графикалық режим үшін) мен баған мен қатар санында (мәтіндік режим үшін) және түсті көрсету тәсілінде болады. Инициализациялау процесі кезінде BIOS бейнеконтроллерді" бастапқы режимін береді. Бұл режим бейнеконтроллерді" түріне байланысты беріледі.

Егер программаға бастапқыдан бөлек режим қажет болса, ол 00h функциясын пайдалануы керек.

Мысалы:

```
mov     AH,    0
mov     AL,    3
int 10h
```

#### ***Курсорды" мөлшері мен пішінін орнату (01h)***

Бұл функция көмегімен курсорды" (ме"зерді") мөлшері мен пішінін (формасын) орната аласыз, курсорды жыпылықтатып қоюы"ызға, мүлдем алып тастауы"ызға болады.

#### ***Курсорды орнату (02h)***

Бұл функция курсорды" экрандағы орнын анықтауды басқарады. Мысалы курсорды алып тастау тәсілдеріні" біріне — оны экран шекарасынан тыс жерге, мысалы анықталмаған 26-қатарға орналастыру сияқты амал жатады.

```
mov     AH,    2
mov     BH,    0      ; 0-ші бейнебет
mov     DH,    5      ; 5-ші қатар
mov     DL,    20     ; 20-ші баған
int 10h
```

#### ***Курсорды" орнын анықтау (03h)***

Программа кез келген уақытта, курсорды" экраныны" қай жерінде тұрғанын анықтай алады. Бұндай анықтау программаға

мысалы курсорды келесі позицияға жылжыту үшін керек болуы мүмкін (о”ға, жоғары, төмен, ағымдағы орнынан 10 символға солға қарай жылжу сияқты).

### ***Бейнежадыны” белсенді бетін та”дау (05h)***

Компьютер бейнежадысында экранны” бір ғана бейнесі емес, бірнешесін еске сақтауы мүмкін. Ол үшін бейнежады (көрінетін бейнелерді еске сақтауға арналған жады, бейнеконтроллерде болады) бірнеше блоктарға бөлінеді. Олар беттер деп аталады.

Бейнежадыны” тек белсенді беті ғана көрініп тұрады. Беттер механизмін пайдалана отырып, программа алдын-ала бейнені белсенді емес бетте дайындап, сосын оны белсенді етуіне болады. Жа”адан пайда болған бет, сол сәтте экранда көрініс береді.

### ***Терезені жоғары/төмен айналдыру (06h, 07h)***

Бұл функциялар көмегімен, та”дап алған экран аймағын, берілген қатар мөлшерінен жоғары не төмен жылжытуға болады. Босатылған орындар бос қатарлармен толтырылады. Бұл функцияларды қолдануды” бір түрі – экранды толық тазалау.

```
mov AH, 06h
mov AL, 0 ; барлық терезені тазарту
mov CH, 0 ; жоғарғы сол бұрышты” қатар нөмірі
mov CL, 0 ; баған нөмірі
mov DH, 24 ; төменгі о” бұрышты” қатар нөмірі
mov DL, 79 ; баған нөмірі
mov BH, 17h ; атрибут (негізгі түс пен символды” түсі )
int 21h
```

### ***Символды және атрибутты оқу/жазу (08h, 09h)***

Бұл функциялар көмегімен бейнежадыдан символды және оны” атрибутын оқуға немесе жазуға болады. Символды жазғанда қайталау санын көрсетуге болады. Қайталау санын көрсететін жазу функциясын, экран аймағын кез келген символмен толтыру да пайдалануға болады.

Символды оқу және жазу курсорды” ағымдағы орнынан басталады.

; символды оқу

```
mov AH, 8h ; символды оқу
mov BH, 0 ; бейнебет
int 10h ; AH регистрде атрибут, AL регистрде
; символ қайтарылады
```

; символды атрибутпен жазу

|     |     |     |                  |
|-----|-----|-----|------------------|
| mov | АН, | 9h  | ; символды жазу  |
| mov | ВН, | 0   | ; бейнебет       |
| mov | ВL, | 07h | ; атрибут        |
| mov | СХ, | 1   | ; символдар саны |
| mov | АL, | 'A' | ; символ         |
| int |     | 10h |                  |

### ***Символды жазу (0Ah)***

Бұл функция символды бейнежадыға жазуға арналған. Атрибутты” жеке мәнін анықтамай жаза беру. Ол үшін ағымдағы атрибут мәні алынады. Жазбаны” еселенген (кратность) түрін де пайдалануға болады.

|     |     |     |                  |
|-----|-----|-----|------------------|
| mov | АН, | 0Ah |                  |
| mov | ВН, | 0   | ; бейнебет       |
| mov | СХ, | 1   | ; символдар саны |
| mov | АL, | 'A' | ; символ         |
| int |     | 10h |                  |

### ***Түстік палитраны та”дау (0Bh)***

Бұл функция бейне тө”ірегіндегі жақтауды” түсін басқаруды реттейді. Жақтауды” түсін мәтіндік режимде, ал оны” негізгі түсін (фонын) графикалық режимде реттеп отырады (EGA, VGA және SVGA бейнеконтроллерлері үшін).

### ***Пикселді (нүктені) жазу/оқу (0Ch, 0Dh)***

Бұл функциялар көмегімен бейнежадыны” та”дап алынған беті үшін, экранны” берілген орнында, нүктені” қажет түсін көрсетіп беруге болады. Немесе осы функция арқылы экрандағы кез-келген нүктені” түсін анықтауға болады (графикалық режимде).

; пикселді экран бетіне шығару

|     |     |     |                            |
|-----|-----|-----|----------------------------|
| mov | АН, | 0Ch | ; пикселді жазу            |
| mov | ВН, | 0   | ; бейнебет                 |
| mov | СХ, | 400 | ; графикалық баған (0-639) |
| mov | DX, | 200 | ; графикалық қатар (0-349) |
| mov | АL, | 14  | ; сары түс                 |
| int |     | 10h |                            |

; бейнебуферден пиксел түсін оқу

|     |     |     |                            |
|-----|-----|-----|----------------------------|
| mov | АН, | 0Dh | ; пикселді оқу             |
| mov | ВН, | 0   | ; бейнебет                 |
| mov | СХ, | 400 | ; графикалық баған (0-639) |

---

```

mov     DX,    200    ; графикалық қатар (0-349)
int     10h         ; AL-де түсті (түсті регистрді” нөмірі)
                        ; қайтарады

```

### ***Символды телетайп режимінде экранға шығару (0Eh)***

Осы функция көмегімен символды экранға жазғаннан со” курсор келесі орынға жылжиды. Бұл кезде BEL (дыбыстық сигнал), бір орынға қайтару, қатарды ауыстыру, қатар басына қайтару сияқты басқару символдары ө”деледі.

```

mov     AH,    0Eh
mov     AL,    'A'
mov     BH,    0
int     10h

```

### ***Ағымдағы бейнережимді анықтау (0Fh)***

Егер программа бейнережимді өзгертсе, онда алды”ғы ескі бейнережимді алдын-ала еске сақтауға және жа”аны орнатуға осы аталған функция мүмкіндік береді. Онан со” бұл функция өз жұмысын бітірер алдында ол ескі режимді қалпына келтіре алады. Сонымен бірге бұл функция экрандағы баған саны мен ағымдағы белсенді күйдегі бейнежады бетіні” нөмірін де қалпына келтіреді (қайтарады).

### ***Бейнеконтроллерге қызмет ету (10h, 11h, 12h)***

Бұл функциялар тек EGA, VGA және SVGA бейнеадаптерлерін пайдаланғанда ғана жұмыс істейді. Олар түстік палитраны орнатады, та”ба генераторын іске қосады (мысалы орыс әріптеріні” қаріптері үшін) және басқа да функцияларды орындайды.

### ***Қатарды экранға шығару (13h)***

АТ және одан да жоғары машиналар класы үшін, EGA, VGA және SVGA дисплейлік адаптерлері бар болғанда бұл функция алдын-ала экранда орны көрсетілген, ұзындығы, атрибуты берілген кез-келген символдар қатарын экранға шығара алады. Сонымен бірге бейне жады бетіні” нөмірінде көрсетуге болады.

```

push    DS
pop      ES
mov     AH,    13h    ; қатарды экранға шығару
mov     AL,    0h     ; BL-да атрибут, курсор жылжымайды
mov     BH,    0      ; бейнебет

```

```
mov     BL,    41h    ; атрибут
mov     CX,    s_len  ; қатарды” ұзындығы
mov     DH,    12     ; қатар
mov     DL,    10     ; баған
mov     BP,    offset line ; қатарды” адресі (ES:BP)
int     10h         ; AL-де түсті (түсті регистрді” нөмірі)
; деректер
line db ‘Телетайп режимде қатарды экран бетіне шығару’
l_len equ $-line ; қатарды” ұзындығы
```

#### 2.1.4. Пернетақтамен жұмыс

INT 16h үзүін ө”деуші пернетақтаға қызмет ететін бірнеше функцияларды орындайды.

Пернетақтаға қызмет ететін функциялар көмегімен батырылған перне кодын орындау немесе пайдаланушыны” кез-келген пернені не пернелер комбинациясын басу моментін күтуді орындау сияқты операцияларды жасауға болады.

Батырылған перне кодалары пернетақтаны” арнайы буферіне орналастырылады. INT 16h үзүіні” функцияларыны” біреуі буфер күйін тексереді, яғни батырылған перне кодасыны” бар жоғын. Егер буфер бос болмаса, ондағы бірінші перне кодасы программаға қайтарылады. Бұл код кейіннен басқа енгізу функциясы арқылы оқылуы мүмкін (күту арқылы енгізу). Бұл функция пайдаланушы кез-келген пернені басқанға дейін программа жұмысын тоқтата тұрады. Содан со” ол программа батырылған перне кодасын қайтарады.

Жоғарыда сипатталған функцияларды” кейбірін пернетақта буферін тазалауға пайдалануға болады: алдымен программа буфер бос па соны тексереді. Егер ол бос болмаса, перне кодасы оқылады. Бұл код ешқайда ауыстырылмайды (жоғалады). Код оқылып болған со”, программа буфер мазмұнын тағы тексереді, осылайша, буфер бос болғанша тексеріле береді.

АТ-дан төмен емес машиналары үшін басқа да функциялар атқарады: пернетақта буферіне символды жазу, тоқтату түруды орнату, ке”ейтілген пернетақтаға қызмет ету.

#### 2.1.5. Дискілік бағыны”қы жүйеге қызмет ету

INT 13h үзуі қатты дискіге және ИМДЖ (НГМД) қызмет етуге арналған. INT 13h үзуіні” көптеген функциялары деректерді оқу және жазуға қатысты барлық операцияларды орындайды. Біз осындай функцияларды” ма”ыздыларын қарастырып өтейік:

***Дискілік жүйені босату (сброс)(00h)***

Бұл функция барлық дискілік жүйені немесе та”дап алынған дискілік құрылғыны бастапқы қалпына келтіруді орындайды. Әдетте құрылғы жұмысыны” бастапқы кезінде пайдаланылады.

***Дискілік жүйені” күйін анықтау (01h)***

Бұл функция алды”ғы операцияны” орындалу нәтижесін тексереді. Егер операция сәтсіз аяқталса, онда бұл функция көмегімен қате кодасын анықтауға болады.

***Диск секторларына оқу / жазу (02h/03h)***

Бұл функциялар көмегімен диск секторларынан компьютерді” жедел жадысын оқу немесе жадыдағы ақпаратты диск секторларына жазуды іске асыруға болады.

Функцияны шақыру алдында құрылғыны”, оны” жолыны”, бастиегіні” нөмірін көрсету керек. Сонымен бірге программа оқылатын немесе жазылатын секторлар санын да көрсетуі керек.

***Секторларды тексеру (04h)***

Функция дискідегі секторлар мазмұнын, циклдік бақылау қосындысын (СКС) пайдалана отырып тексереді.

***INT 13h үзуіні” басқа функциялары***

INT 13h үзуіні” басқа функциялары қатарына: жолды форматтау, дискіні” көрсетілген жолына бастиекті орнату, дискіні тесттеу және оны алдын-ала орнату, контроллер диагностикасын іске қосу және тағы да басқа осы сияқты көптеген функцияларды жатқызамыз.

**2.1.6. Деректерді параллель портқа шығару**

BIOS құрамында INT 17h үзуіні” үш функциясынан тұратын параллель портқа қызмет ететін қарапайым жүйе бар. Бұл функцияларды атап айтсақ: 01h (параллель портты инициализациялау), 02h (параллель порт күйін білу), 00h (параллель портқа байтты шығару). Дербес компьютерге бірнеше

параллель порттарды қосуға болатындықтан, принтерге шығарудан бұрын, порт нөмірін көрсету қажет.

#### **2.1.7. Деректерді жіберуде тізбектелген портқа қызмет ету**

INT 14h үзу функциялары деректерді жіберуде тізбектелген портқа қызмет етеді. Бұл функциялар көмегімен деректерді жіберуді” жылдамдығы мен фоматын беруге болады, порт күйін анықтауға, деректерді жіберуді орындауға болады.

#### **2.1.8. Жүйелік сағаттармен жұмыс**

INT 1Ah үзу функциялары әр компьютерде бар, сағаттарға қызмет етеді. Бұл функциялар көмегімен сіз уақытты, мерзімді тағайындауы”ызға болады. Сағатты” қазіргі күйін білуге болады. Нағыз іс жүзінде өмірдегі сағат күйі, АТ төмен емес класындағы машиналарда бар. Нағыз іс жүзінде сағатты пайдалана отырып, сіз сағатты белгілі бір уақытқа “ояту (будильник)” күйіне қоюы”ызға болады. Содан со” қажет уақытта INT 4Ah үзуі көмегімен дыбыстық сигнал немесе экранда сізге қажет хабарлама болады.

#### **2.1.9. Операциялық жүйені қайталап іске қосу**

INT 14h үзу функцияларын шақыру арқылы (қолданба программа шақырады) операциялық жүйені қайталап іске қосуға болады.

#### **2.1.10. АТ класс машиналарына арналған жүйелік сервис**

INT 15h үзу көмегімен ке”ейтілген пернетақтаға қызмет көрсетіледі, микросекундпен берілетін программалық кідіріс және ке”ейтілген жадыға қызмет ету орындалады.

Сонымен қатар INT 15h үзу функциялары процессорды қорғалған режиміне ауыстырады. Процессорді қайтадан нақтылы режимге ауыстыру бастапқы босату (сброс) сигналы арқылы іске асады. Арифметикалық сопроцессерде де солай болады.

INT 15h үзуіні” C0h функциясы компьютерді” аппараттық құралдарыны” конфигурациясы туралы қосымша мәліметтерді береді.

## 2.2. MS DOS үзулеріне шолу

MS DOS операциялық жүйесі программалық үзулер механизмін пайдалана отырып іске асыратын жүйелік шақырулар жиынтығын, қолта”ба программаларға ұсына алады. Бұл шақырулар қолта”ба программаларды” жүйелік ақпараттан, енгізу/шығару консольдік жүйесінен, файлдық жүйеден программаларды және жадыны басқару жүйесінен мәлімет алуына, компьютерді” шет құрылғылары драйверлеріне қатынауды ұйымдастыруға т.б.с.с. мүмкіндік береді.

MS DOS-ты” негізгі функциялары INT 21h үзулері көмегімен шақырылады. Дегенмен, MS DOS басқа да үзулерді қолданады.

| <i>Үзулер</i> | <i>Сипаттамасы</i>                                                                                    |
|---------------|-------------------------------------------------------------------------------------------------------|
| INT 20h       | Программа жұмысыны” аяқталуы.                                                                         |
| INT 25h       | Секторларды” абсолюттік адресі бойынша дискіден оқу.                                                  |
| INT 26h       | Секторларды” абсолюттік адресі бойынша дискіге жазу.                                                  |
| INT 27h       | Программаны жадыда резидентті түрде қалдыра отырып оны” жұмысын аяқтау.                               |
| INT 28h       | Үзулер MS-DOS қалдырылған, резидентті программалар пайдаланады.                                       |
| INT 2Eh       | MS-DOS командаларын орындау.                                                                          |
| INT 2Fh       | Мультиплексстік үзулер, Print — программасында және басқа да резидентті программаларда пайдаланылады, |
| MS-DOS        | ішкі функциялары мен құрылымдарына жол ашады.                                                         |

INT 21h функцияларын мынандай тораптарға бөлуге болады:

- жүйелік ақпарат алатын;
- символдық енгізу / шығару;
- файлдық жүйемен жұмыс;
- программаларды басқару;
- жадыны басқару;
- құрылғы драйверлерімен байланыс;
- басқа да жүйелік сервис.

Функция нөмірі INT 21h үзулерін шақырғанда АН регистрінде беріліп отырады.

### 2.2.1. Жүйелік ақпарат алу

30h функциясы AX регистріне MS-DOS версиясыны” нөмірін қайтарады. Мысалы, MS-DOS 6.22 версиясы үшін AH регистрі 22h, AL регистрі - 06h те” болады.

Осыған қоса, BH регистрі арқылы функция программаға OEM кодасын (операциялық жүйені өндіруші фирма кодасы) қайтарады (IBM - 00, DEC - 16h, Microsoft - OFFh т.б.с.с.). BL:CH регистрлерінде, функция шақыруынан со”, операциялық жүйені” дистрибутив нөмірі болу керек, бірақ қазіргі MS-DOS версияларында серияны” нөмірі үнемі нөлге те”.

Бұл мәліметтер MS-DOS кейбір версиялары қолдамайтын операциялық жүйе құралдарын талдағанда пайдалануға болады.

2Ah және 2Ch функциялары программаға жүйелік мерзім мен уақытты біліп тұруға мүмкіндік береді.

Ағымдағы диск пен ағымдағы каталог нөмірін қайтаратын функциялар да бар. Бұл функциялар нөмірі –19h және 47h.

2Fh функциясы программаға ағымдағы DTA (Disk Transfer Area) аймағыны” адресін білуге мүмкіндік береді. Бұл аймақ, мысалы, каталогтағы файлдарды іздеуде қолданылады.

PSP (Program Segment Prefix) блогында өте ма”ызды ақпарат болады. Бұл блок жадыда орындалатын программа алдында орналасады. Онда программаны іске қосқанда берілетін параметрлер болады. 62h функциясы ағымдағы PSP блогыны” адресін қайтарады.

Жүйелік ақпарат алуға қажетті функцияларды атап өтейік:

**Функция**

**Сипаттамасы**

**нөмірі**

- |     |                                                                                                                           |
|-----|---------------------------------------------------------------------------------------------------------------------------|
| 35h | Берілген нөмір бойынша ұзу векторыны” мәнін алуға мүмкіндік береді.                                                       |
| 4Dh | Бұл функция көмегімен программаны” аяқталу кодасын білуге болады.                                                         |
| 59h | Қатені” ке”ейтілген кодасын алуға арналған.                                                                               |
| 54h | Дискіге жазуда тексеру пайдаланыла ма, соны анықтайды.                                                                    |
| 33h | Пайдаланушы программа жұмысын <Ctrl+Break> пернелер тіркесі көмегімен тоқтата та ала ма, соны анықтауға мүмкіндік береді. |

## 2.2.2 Символдық енгізу және шығару

Бұл функциялар барлық символдық құрылғылармен (консоль, принтер, тізбектелген порт) жұмыс істеу үшін пайдаланылады. Олар стандартты енгізу - шығару функциялары деп аталады. Стандартты функцияларды пайдаланатын программадағы енгізу және шығаруды мәтіндік файлға ауыстыруға болады.

Символдық енгізу - шығаруды” негізгі функциялары:

| Кода | Мақсаты                        | Сипаты                                                                                                                                                                                                                   |
|------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01h  | Пернетақтадан енгізу           | Стандартты енгізу құрылғысынан символды енгізуді орындайды және оны стандартты шығару құрылғысына шығаруды орындайды. Пайдаланушы программа жұмысын <Ctrl+Break> және <Ctrl+C> пернелер тіркесі көмегімен тоқтата алады. |
| 06h  | Күтпей-ақ пернетақтадан енгізу | Стандартты енгізу құрылғысынан символды күтпей-ақ енгізу және оны стандартты шығару құрылғысына шығаруды орындайды, <Ctrl+Break> және <Ctrl+C> пернелер тіркесі тексерілмейді.                                           |
| 07h  | Тікелей пернетақтадан енгізу   | Символды пернетақтадан енгізу, <Ctrl+Break> және <Ctrl+C> пернелер тіркесі тіркелмейді.                                                                                                                                  |
| 08h  | Пернетақтадан енгізу           | 07h функциясы ұқсас, бірақ <Ctrl+Break> және <Ctrl+C> пернелер тіркесі тексеріледі.                                                                                                                                      |
| 02h  | Символды бейнелеу              | Бейнеленетін символ стандартты шығару құрылғысына жіберіледі.                                                                                                                                                            |
| 09h  | Қатарды бейнелеу               | Стандартты шығару құрылғысы со”ында “\$” белгісі бар қатар жіберіледі.                                                                                                                                                   |
| 03h  | Тізбектелген порттан енгізу    | Тізбектелген порттан символ енгізіледі.                                                                                                                                                                                  |
| 04h  | Тізбектелген портқа шығару     | Символ тізбектелген портқа шығарылады.                                                                                                                                                                                   |
| 05h  | Принтерге шығару               | Символ принтерге шығарылады.                                                                                                                                                                                             |

Кестеде көрсетілгендей, пернетақтадан енгізу үшін бірнеше функцияларды пайдалануға болады. Символды бейнесіз енгізу операциясын парольдер сияқты ақпараттарда пайдалану ы”ғайлы. Егер программа жұмысыны” қисыны пайдаланушы жағынан үзуді

қажет етпесе, онда <Ctrl+C> және <Ctrl+Break> пернелер тіркесін тексерілмейтін функцияларды қолдану ығайлы.

Символдар қатарын шығару үшін 09h функциясын пайдалануға болады, бірақ шығарылатын қатар құрамында “\$” белгісі болмау керек, бұл символ қатар со”ыны” белгісі ретінде пайдаланылады.

### **2.2.3. Файлдық жүйемен жұмыс**

MS DOS программаларға файлдармен және дискілермен жұмыс істеуі үшін, көлемді сервис қызметін ұсына алады. Түгелдей дерлік барлық файлдық операцияларды орындауға әдейі арналған функциялар арқылы атқаруға болады. Тек кейбір жағдайда ғана (мысалы, деректерді көшіруден сақтау) BIOS функцияларына немесе диск контроллер порттарына тікелей қатынауды қолдануға болады.

MS DOS файлдық функциялар көмегімен файлдар және каталогтарды құруға, жоюға, файлдарды ашуға және жабуға, уақытша файлдарды құруға болады. Енгізу, шығару операцияларын буфер арқылы жасауға болады. Программаны” файл мазмұнына тізбектелген және кез-келген түрде де қатынау, мүмкіндігі бар.

### **2.2.4. Программаларды басқару**

MS DOS программаларға басқа программаларды іске қосуға немесе программалық оверлейлерді” орындалуын және жүктелуін іске асыруға мүмкіндік береді. Оны 4Bh функциясы іске асырады.

Программа жұмысын аяқтау үшін MS DOS — ты” арнайы функцияларыны” бірін қолданады. 4Ch функциясы, программа жұмысын аяқтай отырып, операциялық жүйеге бір сан мәнін береді. Ол программаны аяқтау кодасы деп аталады. Бұл санды пакеттік файлда IF ERBORLEVEL командасы арқылы талдауға болады.

Егер бір программа екіншісін іске қосса, онда біріншісі, екінші программаны” аяқтау кодасын 4Dh функция көмегімен ала алады.

Аяқталған программа жедел жадыда қалу үшін (яғни резидентті болуы үшін) ол INT 27h үзуін шақыруы керек немесе 31h функциясын пайдалануы керек.

### **2.2.5. Жадыны басқару**

MS DOS MCB жадысыны” басқару блоктары көмегімен жадыны бөлуді басқарады. Барлық жады әртүрлі көлемдегі үзінділерге бөлінеді. Үзінділер алдында MCB блоктары болады. Ол блоктарда аталған үзінді сипаттамасы болады. (Мысалы, оны” көлемі).

Программа жады аумағын 48h және 49h функциялар көмегімен динамикалық түрде алуына және босатуына болады. Сонымен бірге операциялық жүйені” программаға бөлген үзіндісіні” көлемін де өзгертуге болады. Мұны 4Ah функциясы орындайды.

Himem.sys драйвері программаға адрестік ке”істікте 1-ші мегабайт шегінен жоғары орналасқан, ке”ейтілген жады аумағына қатынауға мүмкіндік береді. Процессор нақтылы немесе виртуалды режимінде ке”ейтілген жады блоктарын адрестеуге шамасы келмегендіктен, оны орындау үшін himem.sys драйверіні” программалық интерфейсін пайдалану қажет.

### **2.2.6. Құрылғы драйверлерімен байланыс**

Драйверлерге қатынау сипаты жасырын түрде өтеді (енгізу-шығару INT 21h үзулері көмегімен). Немесе MS DOS-ты” арнайы функциялары пайдаланылады (кодасы 44h). Бұл функция драйвер мен программа арасындағы басқарушы ақпарат алмасуына арналған.

## **2.3 Қателерді ө”деу**

Программа MS DOS-қа кез-келген операцияны орындау үшін қатынау жасағанда, ол соған сәйкес үзулерді шақырады. Ол үшін, алдын ала процессор регистрлеріне барлық соған қажетті параметрлерді жүктейді.

Егер операция орындалуы қандай-да бір себеппен іске аспаса (параметрлер дұрыс емес, құрылғы қызметі дұрыс емес, сұрау салған операция MS DOS версиясына сәйкес емес т.б.с.с.) онда MS DOS-ты” көптеген функциялары үшін қате белгісі – тасымал жалауы орнатылады. Бұл сәтте AX регистрі арқылы программа қайтарылатын қателер кодаларын атап өтейік.

| Кодасы | Сипаты                                   |
|--------|------------------------------------------|
| 01h    | Функцияны” дұрыс емес кодасы көрсетілген |
| 02h    | Файл табылмаған                          |
| 03h    | Жол табылмаған                           |
| 04h    | Ашық файлдар саны көп                    |
| 05h    | Қатынауға тиым салынған                  |
| 06h    | Дұрыс емес идентификатор                 |
| 07h    | Жадыны басқару блогы бұзылған            |
| 08h    | Жады жеткіліксіз                         |
| 09h    | Жады блогыны” адресі дұрыс емес          |
| 0Ah    | Дұрыс емес орта                          |
| 0Bh    | Дұрыс емес формат                        |
| 0Ch    | Дұрыс емес қатынау                       |
| 0Dh    | Дұрыс емес деректер                      |
| 0Eh    | Қосалқыланған                            |
| 0Fh    | Дискіні көрсетудегі қате                 |
| 10h    | Ағымдағы каталогты жоюға болмайды        |
| 11h    | Басқа құрылғы                            |
| 12h    | Басқа файлдар жоқ                        |

Қателер туралы қосымша мәліметтерді алу үшін 59h функциясы пайдаланылады.

Бұл функцияны шақырғанда ВХ регистрінде қателерді талдау де”гейіні” индикаторы болуы керек. АХ регистріне қайтарылатын қатені” ке”ейтілген кодасынан басқа, программа қателер класын (ВН регистрі), болуы мүмкін іс-әрекеттеріні” кодасын (ВL регистрі), қате болған жерін (СН регистрі) алуы мүмкін.

Өкінішке орай, бұл функция CL, DX, SI, DI, BP, DS, ES регистрлер мазмұнын бұзады. INT 21h үзуіні” 59h функциясы пайдаланылатын программа, бұл регистрлер мазмұнын сақтауды қарастырғаны жөн.

АХ регистріне қайтарылатын қателерді” ке”ейтілген кодасы төменде келтірілген. 1-ден 12h дейінгі кодалар бұдан бұрын келтірілген.

| Кодасы | Сипаты                                          |
|--------|-------------------------------------------------|
| 13h    | Жазудан қорғалған дискіге жазу әрекеті жасалған |
| 14h    | Құрылғыны” белгісіз индентификаторы көрсетілген |
| 15h    | Диск дайын емес                                 |
| 16h    | Команда белгісіз                                |
| 17h    | Тексеруді” циклдық кодасыны” қатесі             |
| 18h    | Сұрату құрылымыны” ұзындығы дұрыс емес          |
| 19h    | Іздеудегі қате                                  |
| 1Ah    | Деректерді тасымалдаушы ортасыны” типі белгісіз |
| 1Bh    | Сектор табылмаған                               |
| 1Ch    | Принтерде қағаз жоқ                             |

|         |                                                          |
|---------|----------------------------------------------------------|
| 1Dh     | Жазу қатесі                                              |
| 1Eh     | Оқу қатесі                                               |
| 1Fh     | Жалпы қате                                               |
| 20h     | Бірігіп пайдаланылған файлға қатынаудағы қате            |
| 21h     | Файлды бұғаттауды” бұзылуы                               |
| 22h     | Дұрыс емес диск                                          |
| 23h     | FCB қатынау жоқ (FCB блоктары өте көп)                   |
| 24h     | Бірігіп пайдаланылған файлға арналған буфер толып кеткен |
| 25h     | Кодалық беттер сәйкес емес                               |
| 26h     | Файл со”ын ө”деудегі қате                                |
| 27h     | Дискіні” толып кетуі                                     |
| 28h-31h | Қосалқыланған                                            |
| 32h     | Тораптық функция іске қосылмаған                         |
| 33h     | Қашықтағы компьютер жауап бермейді                       |
| 34h     | Тораптағы атауларды қайталау                             |
| 35h     | Тораптық атау табылмаған                                 |
| 36h     | Торап бос емес                                           |
| 37h     | Тораптық құрылғы енді жоқ                                |
| 38h     | Командалар тым көп                                       |
| 39h     | Желілік а Тораптық адаптерды” аппаратурасындағы          |
| 3Ah     | Тораптан жауап дұрыс емес                                |
| 3Bh     | Торапта қарастырылмаған қате                             |
| 3Ch     | Сәйкес емес қашықтағы адаптер                            |
| 3Dh     | Баспа кезегі толып кеткен                                |
| 3Eh     | Баспа кезегінде орын аз                                  |
| 3Fh     | Баспа алынып тасталған                                   |
| 40h     | Тораптық атау алынып тасталған                           |
| 41h     | Тораптық ресурсқа қатынауға рұқсат жоқ                   |
| 42h     | Тораптық құрылғыны” типі дұрыс емес                      |
| 43h     | Тораптық атау табылмаған                                 |
| 44h     | Тораптық атаулар тым көп                                 |
| 45h     | Байланыс сеанстары тым көп                               |
| 46h     | Тораптағы ресурстарды бірігіп пайдалану әзірге алып      |
| 47h     | Тораптық сұратуға рұқсат жоқ                             |
| 48h     | Басқа адреске көшу тоқтатылған                           |
| 49h     | Қосалқыланған                                            |
| 50h     | Осындай файл бар                                         |
| 51h     | FCB бөлігіні” қайталануы                                 |
| 52h     | Каталогта дискриптор құру мүмкін емес                    |
| 53h     | INT 24h критикалық қателер ө”деушіні” қатесі             |
| 54h     | Қайта тағайындаулар өте көп                              |
| 55h     | Қосарлы қайта тағайындау                                 |
| 56h     | Дұрыс емес пароль                                        |
| 57h     | Параметр дұрыс емес                                      |
| 58h     | Тораптағы деректерді жазудағы қате                       |
| 59h     | Торапта мұндай функция жоқ                               |
| 5Ah     | Жүйені” қажетті компоненті орнатылмаған.                 |

ВН регистрінде берілетін қателер класы өз құрамында сізді” қате”ізді жөндеуге мүмкіндігін беретін мәліметті сақтайды.

| Қателер класы | Сипаты                                       |
|---------------|----------------------------------------------|
| 1             | FCB блоктары, жады, т.б.с.с. ресурстар аз    |
| 2             | Уақытша жағдай, операцияны тағы бір қайталау |
| 3             | Қатынауға құқығы жоқ                         |
| 4             | MS-DOS-тың ішкі қатесі                       |
| 5             | Аппаратура қатесі                            |
| 6             | MS-DOS-тың жүйелік қатесі                    |
| 7             | Қолданбалы программадағы қате                |
| 8             | Файл немесе объект табылмады                 |
| 9             | Файлды немесе объектің форматы дұрыс емес    |
| 10            | Файл немесе объект бұғатталған               |
| 11            | Деректер тасымалдаушысының қатесі            |
| 12            | Осындай файл немесе объект бар               |
| 13            | Басқа қателер                                |

BL регистрі арқылы берілетін іске асуға тиісті шаралар кодасы, сіздің программаңыздың қателерді дұрыс өңдеуіне көмегін тигізеді. Осы кодалар тізімін келтірейік.

| Кода | Ұсынылған іс-әрекеттер                                                                                                                                                                                      |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Операцияны біраздан соң қайтала. Пайдаланушыдан операцияны қайталауға бола ма, не программа жұмысын аяқтауға бола ма деп сұрауға болады.                                                                    |
| 2    | Алдыңғы операцияны біраз үзілістен соң қайтала. Егер қате жоғалып кетпесе, онда пайдаланушыдан әрі қарай тоса ма, әлде программа жұмысын аяқтауға бола ма деп сұрауға болады.                               |
| 3    | Егер пайдаланушы MS-DOS-қа арналған кейбір деректерді енгізсе, одан бұл деректерді тағы да енгізуін сұрау қажет (мысалы ол дискіні дұрыс емес идентификаторын немесе файлға жолды дұрыс көрсетпеуі мүмкін). |
| 4    | Әдетте қолданатын барлық аяқтау іс-әрекеттерін орындай отырып (файлдарды жабу, буферлерді дискіге жазып кету, жады блоктарын босату, т.б.с.с) апатты түрде қолданбалы программа жұмысын аяқтау.             |
| 5    | Аяқтау іс-әрекетін орындамай, программадан бірден шығу. Жүйе болжай алмастай жағдайда.                                                                                                                      |
| 6    | Қатені еске алмаса да болады.                                                                                                                                                                               |
| 7    | Пайдаланушы қажет талаптарды орындаған соң (дискетаны орнатады т.б.с.с) операцияны қайталау керек.                                                                                                          |

СН регистрінде қателерді орналасуы туралы мәліметтер беріледі. Осы қателер тізімі төмендегідей болады.

| Кода | Қате болған жері                                                                  |
|------|-----------------------------------------------------------------------------------|
| 1    | Жүйе қатені” қайда болғанын білмейді                                              |
| 2    | Жекелеген блоктар арасында деректер алмасатын құрылғыда (диск не магниттік лента) |
| 3    | Қате торапқа байланысты                                                           |
| 4    | Символдық құралғыда, мысалы принтерде                                             |
| 5    | Қате жедел жадыға байланысты                                                      |

Егер сізді” программа”ыз ассемблер тілінде құралса, онда үзулер арқылы MS-DOS-қа қатынарудан кейін тасымал жалауыны” мәнін тексеру қажет:

```
int 21h
```

```
jc error
```

MS-DOS - қателерді ө”дейтін тағы бір құралы бар: **Критикалық қателерді ө”деуші**. Бұл модуль MS-DOS-тан құрылғы драйверінен қате туралы хабар түскен со” шақырылады. Модуль экранға, бізге өте таныс хабар береді:

**Abort, Retry, Fail?**

Бұл хабар әдетте сіз дискетаны қоюды ұнатқанда немесе қосылмаған принтер жағдайында басу операциясын жүргізгенде пайда болады. Қолданбалы программалар өздеріні” критикалық қателерді ө”деуші модульдерін, стандартты модуль орнына іске қосуына болады.

## 2.4. Жүйені” пернетақтамен өзара әрекеттесуі

Пернетақта үш түрлі кодалармен жұмыс істейді. Олар: скан-кодалары, ASCII-кодалары және ке”ейтілген ASCII-кодалары. Скан-кодасы — пернені” нөмірін, ASCII-кодасы - символды” (экран бетіне бейнелеуге болады) кодасын, ке”ейтілген ASCII-кодасы — (экран бетіне бейнеленбейді) функциялық пернелерін, курсорды басқару пернелерін немесе бірнеше пернені” тіркестерін көрсетіп тұрады.

Пернетақтаны” жұмысын пернетақта контроллері басқарады. Контроллерді” қызметі: басылған пернені білу және басылған пернені” кодасын өз шығару регистріне (60h портқа) жіберу. Портқа түскен пернені” кодасы — скан-кодасы деп аталады. Әр пернені” екі скан-кодасы болады, кодаларыны” айырмасы 80h. Скан-кодаларыны” біреуі (кіші) контроллермен 60h портқа басу кезінде, ал екіншісі (үлкені) пернені босату кезінде жіберіледі.

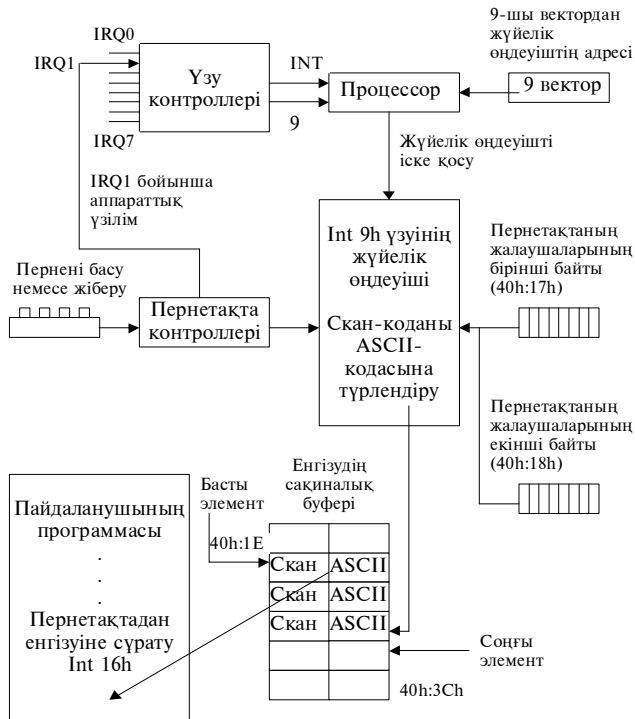
Кезкелген пернені” басылуы немесе босатылуы аппараттық үзу сигналын шақырады. Процессор сигналмен бірге үзу векторыны” нөмірін алады. Үзу ө”деуіші 9h векторы арқылы шақырылады, сондықтан оны INT 09h программасы деп атайды. Үзу ө”дейтін программаны” (ҮӨП) адресі былай есептелінеді:

$$9h(\text{үзу нөмірі}) * 4 = 24h$$

24h адресі бойынша ҮӨП-ны” ығысу адресі, ал 26h адресі бойынша ҮӨП-ны” сегмент адресі болады.

INT 09h программасы 60h порттан басқа жедел жадыны” екі аймағымен: енгізуді” сақиналық буферімен (40h:1Eh бастап 40h:3Dh дейін адрестер бойынша орналасқан) және пернетақтаны” жалаушаларыны” екі байтымен (40h:17h және 40h:18h адрестер бойынша орналасқан) жұмыс істейді. Со”ғы екі байтта басқару пернелеріні” (Shift, Caps Lock, Num Lock және т.с.с.) күйі беріледі.

Кез келген пернені басқанда INT 09h программасы 60h порттан басу скан-кодасын есептейді және оны” мәнін талдайды. Егер скан-кодасы басқару пернесінде болса, онда пернетақта жалаушаларыны” бірінші байтында (2.1-кесте) басылған пернеге сәйкес биті (разряды) орнатылады. Жалаушаларды” күйлері пернелер басылып тұрғанша сақталып тұрады. Егер басқару перне жіберілсе INT 09h программасы босату скан-кодасын алып, оған сәйкес жалаушалар байтыны” битін босатады.



2.1-сурет. Пернетақтаның жүйемен өзара әрекеттесу процесі

Басқару пернелер туралы қосымша ақпаратты жалаушаларының” екінші байтынан (2.2-кесте) алуға болады.

2.1-кесте  
Пернетақта жалаушаларын бірінші байтты” биттеріні” тағайындауы (40h:17h)

| Бит нөмірі | Битті” тағайындауы                  |
|------------|-------------------------------------|
| 0          | Shift o” пернесі басылып тұр        |
| 1          | Shift сол пернесі басылып тұр       |
| 2          | Ctrl кез келген пернесі басылып тұр |
| 3          | Alt кез келген пернесі басылып тұр  |
| 4          | Scroll Lock режимі қосылып тұр      |
| 5          | Num Lock режимі қосылып тұр         |
| 6          | Caps Lock режимі қосылып тұр        |
| 7          | Ins режимі қосылып тұр              |

**Пернетақта жалаушаларын екінші байтты”  
биттеріні” тағайындауы (40h:17h)**

| <b>Бит нөмірі</b> | <b>Битті” тағайындауы</b>       |
|-------------------|---------------------------------|
| 0                 | Ctrl сол пернесі басылып тұр    |
| 1                 | Alt сол пернесі басылып тұр     |
| 2                 | Ctrl о” пернесі басылып тұр     |
| 3                 | Alt о” пернесі басылып тұр      |
| 4                 | Scroll Lock пернесі басылып тұр |
| 5                 | Num Lock пернесі басылып тұр    |
| 6                 | Caps Lock пернесі басылып тұр   |
| 7                 | SysReg пернесі басылып тұр      |

Жай пернені басқанда INT 09h программасы 60h порттан оны” басу скан-кодасын есептейді және скан-кодларды ASCII-коддаларына аудару кестесі бойынша екі байтты кодасына қалыптастырады (үлкен байтта – скан-кодасы, кіші байтта - ASCII-кодасы). Аудару нәтижесінде алынған екі байттық кода енгізуді” сақиналық буферіне (көлемі 16 сөз) жібереді.

Символдар кодалары буферге қалай түссе, буферден сондай тәртіппен шығады. Буферді” жағдайын екі көрсеткіш бақылап отырады. Со”ғы көрсеткіште (40h:1Ch адресі бойынша сөз) бірінші бос ұяшықты” адресі сақталады. Ал бастапқы көрсеткіште (40h:1Ah) пернетақтадан алынған е” ескі коданы” адресі сақталады. Буфер бос болған жағдайда көрсеткіштерді” екеу де буферді” бірінші бос ұяшығына сілтеп тұрады.

Екі байттық кодасы буферге со”ғы көрсеткіш адресі бойынша келіп түседі де, содан со” адрес 2-ге ұлғаяды.

Пернені” басу кодасын алу үшін орындалатын программа BIOS-ты” (int 16h) немесе DOS-ты” (int 21h) үзулерді” функцияларын шақыруы тиіс. Жүйелік программалар пернетақтаны” драйвері көмегімен басты көрсеткіште, адресі бойынша сақиналық буферінен ұяшықты” мазмұнын есептейді, содан со” адресі екіге арттырады.

Кодаларды енгізу процесінде со”ғы көрсеткіш буферді” со”ына жетеді (40h:3Ch адресі). Енді кезек кодасы келгенде көрсеткішті” адресі буфер ұзындығына азаяды, сонымен көрсеткіш буферді” басына қайтып келеді. Осындай әрекеттер басты көрсеткішпен де орындалады. Егер екі көрсеткіштерді” адрестері бірдей болса, онда буфер бос болғаны.

### 3. ФАЙЛДЫҚ ЖҮЙЕ

#### 3.1. Файлдарға, каталогтарға, дискілерге қызмет көрсетуді” жүйелік құралдары

IBM PC машиналарында магнитті дискілерге қатынасуды” екі де”гейі қаралған. Төменгі де”гейде жұмыс істегенде пайдаланушы BIOS INT 13h үзуіні” көмегімен дискжетектерге қатынаса алады. Бұл де”гейге тән операциялар қатарына секторға жазу немесе одан оқу, бастиектерді бағыттау, жолшықты пішімдеу (форматирование) сияқты операциялар жатады. DOS файлдық жүйесі қолданылмайды; керек ақпарат файлды” атымен емес, бетті”, цилиндрді” және секторды” нөмірлері арқылы табылады.

Жоғарғы де”гей DOS INT 21h үзуі бойынша жүзеге асады. Оны” құрамында файлдық құрылымға қызмет ететін функциялар болады. DOS-ты” файлдық жүйесі ұғымына логикалық диск, каталог, файл деген ұғымдар кіреді.

Саны көп әртүрлі файлдармен жұмыс істеуге ы”ғайлы болу үшін DOS-та каталогтарды” бұтақталған құрылымы пайдаланылады. Каталог дегеніміз, келесі төменгі де”гейдегі бүкіл ішкікаталогтар тізімін және оған кіретін файлдарды бойына жинақтаған көлемі аса үлкен емес файл.

Әрбір ішкікаталогқа немесе файлға өлшемі 32 байт жазба бөлінеді. Оған DOS файл туралы ақпаратты жазады: аты, дискідегі бастапқы адрес (кластер нөмірі), құрылған уақыты мен мезгілі, байт бойынша ұзындығы, файл атрибуттары. Әр каталогта, төменгі де”гейдегі каталогтар мен файлдар жазбаларынан басқа тағы екі түрлі жазба болады: өзі туралы және аталық каталог туралы. Каталогты” жазба пішімі 3.1-кестеде келтірілген.

3.1-кесте

Каталог жазбаны” пішімі

| Ығысу | Байт саны | Мазмұны                                  |
|-------|-----------|------------------------------------------|
| 00h   | 8         | ASCII кодаларында файл аты               |
| 08h   | 3         | ASCII кодаларында файл атын ке”ейту      |
| 0Bh   | 1         | Файл атрибуттарды” байты                 |
| 0Ch   | 10        | Қосалқыланған                            |
| 16h   | 2         | Файлды құру немесе со”ғы өзгерту уақыты  |
| 18h   | 2         | Файлды құру немесе со”ғы өзгерту мезгілі |
| 1Ah   | 2         | Дискіде файл басталатын кластер нөмірі   |
| 1Ch   | 4         | Файлды” байт бойынша нақты ұзындығы      |

Файлды” атрибуттары (оны” ішінде каталог файлыны”) арнайы бөлінген атрибуттар байтында сақталады да төмендегі мәндерге ие болады :

01h - тек оқу үшін керекті файл. Файлды өзгертуге немесе жоюға тыйым салынады.

02h - жасырын файл (DIR командасымен немесе DOS-ты” бірнеше қызметтерімен шығара алмайтын).

04h - жүйелік файл (әдетте IO.SYS және MSDOS.SYS файлдар жүйелік және сонымен бірге жасырын болып жарияланады).

08h - файл туралы жазу томны” белгісін білдіреді. Мұндай жазу (біреуі барлық томға) тек бастапқы каталогта ғана болады.

10h - файл каталогты білдіреді.

20h - файлды өзгертуден кейін немесе құрылғаннан кейін архивтендірілген жоқ, сәйкесінше, архивтеу қажет (архивтеу атрибуты).

Файлды” бір мезгілде бірнеше атрибуттары болуы мүмкін. Мысалы: 28h - том та”басы, архивтелмеген, 21h - файл өшіруден және өзгертуден қорғалған, архивтелмеген, 23h - жасырын файл, файл өшіруден және өзгертуден қорғалған, архивтелмеген.

Файлды” құрылған уақыты мен мезгілі, осы файлмен жұмыс істейтін программа жұмысын аяқтап, файл жабылған со”, каталог жазбасына орналастырылады. Бұл өрістер пішімі 3.2 және 3.3 - кестелерде келтірілген.

### 3.2-кесте

#### Файлды” құрылған уақыты өрісіні” пішімі

| Разрядтар | Тағайындау                                                                                       |
|-----------|--------------------------------------------------------------------------------------------------|
| 04...00   | Екілік түріндегі екі секундтық аралықтарды” саны (1-ден 29 дейін, яғни 0 сек- тан 58 сек. дейін) |
| 10...05   | Екілік түріндегі минут саны (0 ден 59 дейін)                                                     |
| 15...11   | Екілік түріндегі сағат саны (0 ден 23 дейін)                                                     |

Файл басталатын кластер нөмірі, дискідегі файл бастамасын табуға мүмкіндік береді; одан кейінгі кластерлер жайлы ақпарат (FAT) файлдар орналасуы кестесінде болады. Каталог жазбасыны” со”ғы төрт байтында.

## Файлды” құрылған мезгіл өрісіні” пішімі

| Разрядтар | Тағайындау                                     |
|-----------|------------------------------------------------|
| 04...00   | Ай күні (1- ден 31- дейін)                     |
| 08...05   | Ай -(1 ден 12- дейін)                          |
| 15...09   | 1980 жылдан бастап жыл саны (0- ден -23 дейін) |

байтпен берілген файлды” ұзындығы сақталады. Егер модификациялау процесі нәтижесінде файл өлшемі өзгерсе, онда DOS аталған өріс мәнін өзгертеді.

DOS жа”а файл құрғанда, дискіде бос орынды өзі іздеп табады да, оны жа”а файлға береді, сонымен бірге каталогтағы соған сәйкес жазбаны толтырады.

Каталогта файлды” логикалық ұзындығы көрсетіледі.

Файлды жазу немесе оқу процесінде дискіні” контроллері беретін ақпаратты” е” аз мөлшері сектор (512 байт) болады (BIOS программалары осы секторлармен жұмыс істейді), бірақ файлдық жүйе дискіде оларға бүтін кластерлермен орын белгілейді.

Иілгіш дискідегі кластерді” өлшемі екі сектор (1кбайт); ал қатты дискіде кластерге 4-8 сектор сияды. Осымен, файлды” (ондағы берілгендер тек бірнеше байт йемденсе де) е” аз өлшемі бір кластер болады. Бірақ каталогты” жазуында физикалық емес, логикалық ұзындығы көрсетіледі.

Файлдармен жұмыс істеуді” әдістері әрбір файлды” дискідегі йемденетін аумағына байланысты болады. Мұндай жүйе, біріншіден, жұмыс кезінде файлға берілгендерді бірнеше дүркін жазуға мүрсат береді (файлды” ұзаруын қосқанда) екіншіден, дискіні” фрагментациясындағы проблемаларды түсіреді, өйткені е” кішкентай және бос аумақтар жа”а файлды орналастыруға пайдаланулары мүмкін. Бірақ, өте фрагменттелген файл оқу не жазу үшін көп уақыт керек ететінін ескеру керек, ал ол программаны” орындалу жылдамдығын төмендетеді.

Файлдармен операциялар орындауды” екі әдісі бар. (FCB) файлдарды басқару блогын және (handle) файлдар дескрипторларын пайдалану арқылы. Файлдарды басқару блогы дегеніміз файл туралы ақпаратты бойына жинақтаған 37 байттық кесте, ол мынандай ақпаратты береді: аты және ке”ейтуі, өлшемі, файлдағы жазбаларды” ұзындығы, ағымдағы жазбаны” нөмірі және т.с. Бұл кесте жадыда болады. Файлды орындаған кезде FCB-ны толтыру керек және керекті функцияны шақыру керек.

FCB-ні пайдалану DOS-ты” алғашқы нұсқаларында көп кездескен. FCB құрылымдары және DOS-” сәйкес функциялары бұтақтар теріздес каталогты қолдамайды, сондықтан олар тек ағымдағы дискіні”, ағымдағы каталогтағы файлдарымен жұмыс істей алады. Қазіргі кезде FCB өте сирек қолданылады.

Файлдармен операциялар орындауды” басқа әдісі дескрипторларды (файлдық сипаттауыштарды) пайдалануды жүзеге асырады. Оларды ашық файлдар нөмірлері ретінде қарауға болады.

Файлға қатынасуды” жолы төмендегідей операцияларға бөлінеді:

- көрсетілген каталогта берілген атпен файл құру немесе ертерек құрылған болса, файлды ашу;
- файлға жазу немесе файлдан оқу;
- файлды жабу.

Көп жағдайда файлдармен жұмыс оларды ашудан басталады. DOS файлды ашып, оған арнайы жүйелік кестені” кезекті бос элементін (файлдық сипаттау блогты) тағайындайды. Бұл кесте (System File Table, SFT) ашық файлдар кестесі деп аталады және оны” көлемі CONFIG.SYS файлыны” FILES деген директивасымен беріледі.

Дискіні” каталогтар жүйесінде файлды ашу туралы жазбаны тапқан со”, DOS оған бөлінген элементті SFT файл сипаттау блогы жазады, яғни оған файлды” негізгі сипаттамаларын (аты, ұзындығы, атрибуттары, құру уақыты және мезгілі, бастапқы кластері, дискідегі каталог жазбасыны” физикалық адресі, т.б.) жазады.

Ақпаратты” бір бөлігі SFT элементіне каталог жазбасынан жазылады, басқа бөлігі (мысалы дискіні” блок параметрлеріні” нұсқағышы, онда дискіні” физикалық сипаттамалары сақталады) DOS өзі береді. Файл сипаттау блогыны” ма”ызды элементтері қатарына қос сөздік ұяшық жатады. Онда файлды” басына қажет кезекті жазу немесе оқу операциясы басталатын, файл нұсқағышы сақталатын байт нөмірі болады. Нұсқағышты” болуы файлға тура қатынауды ұйымдастыруға мүмкіндік береді. Яғни оқу мен жазу файлды” кез келген жерінен мүмкін болады. Бөлінген блок нөміріне сілтемені SFT-да DOS программаға дескриптор түрінде қайтарады.

Ашық файлға қатынау (оқу, жазу, файл сипаттамаларын өзгерту т.б.с.) берілген дескриптор көмегімен іске асады.

Ашылмаған файлда дескриптор болмайды және онымен жұмыс істемейді. Ашық файлмен жұмыс істеу барысында DOS ақпаратты SFT блогында өзгертеді. SFT мазмұны үнемі файлды” ағымдағы күйін білдіріп тұрады.

Файлмен жұмыс істеп болған со” оны жабу керек. Жабу процесі кезінде DOS буферлер мазмұнын дискіге жазуы және каталог жазбасын өзгертуі, SFT-дағы файлды сипаттау блогымен бірге соған бекітілген дескрипторды босатуы іске асады.

CONFIG.SYS файлыны” BUFFERS директивасымен саны анықталатын DOS буферлері файлмен жұмысты тездетуге қызмет етеді. DOS файлдан деректерді оқығанда, оны программаға жіберумен бірге, оларды өзіні” ішкі буферлерінде сақтайды. Бұл деректерге қайтадан қатынау кезінде, олар яғни дискіден емес DOS-ты” ішкі буферінен оқылады. Файлды жабу процесі кезінде DOS буферлерін дискіге жазып тастау арқылы дискідегі файлдарды” физикалық түрдегі жа”аруы өтеді.

Файлмен жұмыс болып жатқан кезде, файлдар сипаттамасы туралы мәліметтерді” жа”аруы тек SFT файлды сипаттау блогында өтеді. Файл жабылғанда файлдар сипаттамасыны” өзгеруі SFT-дан каталог жазбасына көшіріліп жазылады және оған бөлінген файлды сипаттау блогы өзіні” дескрипторымен бірге босайды.

Программа аяқталу кезінде программадан барлық ашылған файлдарды” автоматты түрде жабылуы орындалады.

DOS-ты” дескрипторлар арқылы болатын оқу және жазу файлдық функцияларыны” ерекшелігі мынада: оларды компьютерды” стандартты құрылғылары арқылы енгізу-шығаруға пайдалануға болады. Стандартты құрылғыларымен жұмыс істеу үшін DOS арнайы анықталған бес дескрипторды береді.

- 0 - стандартты енгізу (CON)
- 1 - стандартты шығару (CON)
- 2 - стандартты қате (CON)
- 3 - стандартты көмекші порт (AUX)
- 4 - стандартты принтер (PRN)

Сонымен терминалмен, принтермен немесе тізбекті портпен жұмыс істеген кезде жа”а дескрипторларды ашуды” керегі жоқ. Пернетақтадан енгізу 0 дескрипторы арқылы, экранға шығару - 1 не 2 дескрипторы арқылы, принтерге шығару - 4 дескрипторы арқылы жүзеге асырылады.

1 және 2 дескрипторларыны” айырмашылығы: стандартты шығаруды (енгізуді), DOS құралдары көмегімен, кез-келген

құрылғыға немесе файлға бағыттап жіберуге болады. Ал стандартты қате үнемі экранмен байланысты. Әдетте, 2 дескриптор экранға апатты және диагностика хабарларды шығарып тұруға пайдаланылады.

Енгізу немесе шығару программасын басқа жаққа бағыттауды `command.com` командалық процессоры жүзеге асырады. Мысалы, егер `PRIM.EXE` программасында деректерді енгізу (0) стандартты енгізу дескрипторы арқылы орындалса, ал деректерді шығару (1) стандартты шығару дескрипторы арқылы орындалады. Онда программаны келесі командамен орындау:

`PRIM.EXE < FILE_IN.TXT > FILE_OUT.TXT`

деректерді `FILE_IN.TXT` файлынан енгізіп, ал шығару деректерін экранға емес `FILE_OUT.TXT` файлына шығаруды іске асырады.

`DOS-ы` көптеген қызметтері `ы` файллы болу үшін мағыналы топтарға бөлу керек.

**1. Файлды құру, ашу және жабу:**

3Ch - файл құру;

5Ah - уақытша файл құру;

5Bh - жа"а файл құру;

3Dh - файл ашу;

3Eh - файл жабу;

68h - файлды дискіге жазып салу;

41h - файлды жою.

**2. Деректерді жазу және оқу:**

42h - нұсқағышты қою;

3Fh - файлдан не құрылғыдан оқу;

40h - файлға не құрылғыға жазу.

**3. Файлды" сипаттамаларын өзгерту:**

43h - файл атрибуттарын қою не алу;

56h - файл атын өзгерту;

57h - файл құрылған уақыты мен мезгілін алу не қою.

**4. Файлды іздеу:**

1Ah - (DTA) деректерді беру аймағыны" адресін орнату;

2Fh - (DTA) деректерді беру аймағыны" адресін алу;

4Eh - бірінші файлды табу;

4Fh - келесі файлды табу.

**5. Каталогтармен операциялар:**

39h - каталогты құру;

3Ah - каталогты жою;

3Bh - ағымдағы каталогты ауыстыру;

47h - ағымдағы каталогты алу.

## 6. Дискілермен операциялар:

19h - ағымдағы дискіні алу;

0Eh - ағымдағы дискіні ауыстыру;

36h - диск туралы мағлұмат алу.

Файлдармен жұмыс кезінде сәтті орындалған операция CF жалауына 0 орнатады. Егер операция орындалмаса, онда CF жалауына 1 орнатылады, ал AX регистріне қате кодасы қойылады.

Сонымен, осы қызметтерді” көмегімен “нағыз” файлды” да, томны” белгісінде құруға болады (дискіні” түрі каталогында). AX регистрінде құрылған файлды” дескрипторы қайтарылады. 3Ch, 5Bh қызметтеріні” айырмашылығы файл болған кезде білінеді.

**3Ch және 5Bh функциялары** файлды берілген спецификациямен құруға мүмкіндік береді. Файл спецификациясы (файлға көрсететін жолы мен файл аты ASCIIZ-қатар түрінде) екілік нольмен бітетін символдық қатар түрінде көрсетіледі. Бұл қатарды” адресі DS:DX регистріне енгізіледі. Бұл функцияларды” айырмашылығы: 3Ch функциясы бар файлды жояды да және осы атпен жа”а файл құрады, ал 5Bh функциясы - егер осындай аты бар файл болса, онда CF=1 болып бітеді.

Мысалы:

; Деректер сегменті

fname DB 'F1.TXT',0 ; файл аты

handle DW ? ; дескрипторға арналған ұяшық

; Код сегменті

mov AH, 3Ch ; файл құру функциясы

mov CX, 00 ; әдеттегі файл - атрибутсыз

lea DX, fname ; файл атыны” адресі

int 21h

jc ERR ; қате бойынша өту

mov handle, AX ; дескрипторды сақтау

Бұл мысалда CX регистріне файл атрибуты енгізіледі. Келесі атрибуттарды қолдануға болады: 01h - тек оқуға арналған файл, 02h - жасырылған файл, 04h - жүйелік файл, 08h - том белгісі, 10h - файл өзімен каталогты көрсетеді, 20h - архивтеу атрибуты.

**5Ah** қызметі уақытша файл құруға арналады, ол файл атты жүйе береді. DS:DX регистрлерінде ASCIIZ қатарындағы файлға

баратын адрес көрсетіледі. Ал оны” аяғында 13 бос байт қарастырылады. DOS екілік нөлмен аяқталатын, құрылған файлды” атын көрсетеді. Керек жағдайда файлға томны” белгісінен басқа кез-келген атрибуттарды беруге болады. Әдетте уақытша файлдар программаны” со”ында ажырайды. Бұл файлға жазу үшін AX регистріндегі 5Ah қызметіні” дескрипторын пайдаланады.

**3Dh** қызметі бар файлды ашуға мүрсат береді. DS:DX регистрлерінде файлды” спецификациясы беріледі (файлды” жолы мен ке”еюі ASCII қатарында; AL – қатынауды” режимі: 0-оқу, 1-жазу, 2-оқу және жазу). Бұдан былай файлды оқу мен жазу AX регистріндегі дескрипторды” көмегімен жүзеге асырылады.

**3Dh функциясы** - бар файлды ашуға мүмкіндік береді.

Мысалы:

```
mov      AH,    3DH    ; файлды ашу функциясы
mov      AH,    2      ; енгізу - шығару үшін
lea      DX,    fname ;
int      21h
mov      handle, AX
```

Әрбір ашық файл үшін DOS көрсеткіш дайындайды, ол-байтты” нөмірі. Құрылған файлды” көрсеткіші жүйемен позицияланады, оқу және жазу қызметтері оны оқылған не жазылған байттарды” санына ауыстырады. Сонымен қайталанған оқу және жазу қызметі файлға жол ашады. Файлды” орнына шығу үшін арнайы қызмет бар. (Бұл үшін AL=0 болу керек), файлды” со”ы (AL=2) немесе көрсеткішті” со”ғы орны (AL=1). Көрсеткішті” ауыстыру мәні CX (үлкен жартысы) және DX регистрлеріне (кіші жартысы) енгізіледі.

**42h функциясы** файлды” кез келген орнына тура қатынауды ұйымдастыру үшін қолданылады. Нұсқағышты файлды” со”ына (AL=2) орнатып файлды” ұзындығын анықтауға болады. Функциядан қайтқанда DX (үлкен жартысы) және AX (кіші жартысы) регистрлерде файлды” ұзындығы болады. Мысалы:

```
mov      BX,    handle
mov      AH,    42h    ; нұсқағышты орнату
mov      AL,    2      ; файлды” со”ына
mov      CX,    0

mov      DX,    0
```

```

int      21h
mov      word ptr fsize, AX
mov      word ptr fsize+2, DX
; деректер
fsize    dd      0

```

3Fh, 40h қызметтері файлды не құрылғыны оқу не жазу үшін қолданылады. Шақыру алдында BX регистріне дескриптор орналасады, CX регистріне оқылған не жазылған байттарды” саны, ал DS:DX регистріне-буферді” адресі орналасады.

**3Fh функциясы** файлдан немесе құрылғыдан оқуға пайдаланылады.

Мысалы:

```

mov      AH,      3Fh      ; оқу функциясы
mov      BX,      handle; дескрипторды орнату
mov      CX,      80      ; қанша оқу қажет
lea      DX,      buf      ; және қайда
int      21h
mov      CX,      AX      ; факты бойынша қанша оқылды

```

**40h функциясы** файлға немесе құрылғыға жазу үшін пайдаланылады.

Мысалы:

```

mov      AH,      40h      ; жазу функциясы
mov      BX,      handle; дескриптор орнату
mov      CX,      80      ; қанша жазу қажет
lea      DX,      buf      ; және қайда
int      21h

```

**3EH функциясы** файлды жабу үшін пайдаланылады. Бұл операция каталогты және FAT таблицасын түзету үшін қажет.

Мысалы:

```

mov      AH,      3Eh      ; жабу функциясы
mov      BX,      handle
mov      21h

```

**41h функциясы** файлды жою үшін пайдаланылады. (Тек "оқу үшін ғана" деген файл атрибутынан басқа) .

```

mov      AH,      41h      ; жою функциясы
lea      DX,      fname      ; файл атыны” адресі
int      21h      ; DOS - ті шақыру

```

**43h функциясы** файлды” атрибутын алу үшін (AL=0), немесе орнату үшін (AL=1) пайдаланылады.

Мысалы:

```
mov    AH,    43h    ; атрибут жұмысыны” функциясы
mov    AL,    1      ; атрибуттар орнату
mov    CX,    1      ; "оқу үшін ғана"
mov    DX,    offset fname ; файл атыны” адресі
int     21h
```

**56h функциясы** файлды” атын өзгерту үшін пайдаланылады.

Мысалы:

; сегменттік регистрді ES деректер сегментіне баптау.

```
push    DS
pop     ES
```

; файлды” атын өзгерту

```
mov     AH,    56h    ; ат өзгерту функциясы
mov     DX,    offset oldname ; ескі атыны” адресі
mov     DI,    offset newname ; жа”а атыны” адресі
int     21h
```

; деректер сегментіне

```
oldname db 'F1.TXT',0
newname db 'NEWF.TXT',0
```

**57h функциясы** файлды” құрылған уақытын алу үшін (AL=0) және күнін орнату үшін (AL=1) пайдаланылады.

Уақыт CX регистріне жазылады, сосын мына формуламен есептелінеді.

$CX = \text{сағат} * 2048 + \text{минуттар} * 32 + \text{секунді} / 2$

Ал күн болса DX регистріне жазылады және мына формуламен есептелінеді.

$DX = (\text{жыл} - 1980) * 512 + \text{ай} * 32 + \text{күн}$

Мысалы:

; файлды” құрылған күні мен уақытын өзгерту

```
mov     AH,    57h    ; уақыт / күн функциясы
mov     AL,    1      ; уақытты / күнді орнату
mov     BX,    HANDLE ; дескрипторды орнату
mov     CX,    0      ; CX - тазалау
or      CX,    sec     ; секундты қосу
or      CX,    min     ; минутты қосу
or      CX,    hour    ; сағатты қосу
xor     DX,    DX      ; DX - тазалау
or      DX,    day     ; күнді қосу
```

|     |     |      |              |
|-----|-----|------|--------------|
| or  | DX, | mon  | ; айды қосу  |
| or  | DX, | year | ; жылды қосу |
| int | 21h |      |              |

; деректер сегментіне

|      |    |         |                      |
|------|----|---------|----------------------|
| sec  | dw | 6/2     | ; 6 секунд           |
| min  | dw | 15*32   | ; 15 минут           |
| hour | dw | 16*2048 | ; 16 сағат           |
| day  | dw | 25      | ; 25 күннің саны     |
| mon  | dw | 3*32    | ; наурыз             |
| year | dw | 15*512  | ; 15 жыл (1995-1980) |

Файлмен жұмыс бірнеше кезе”нен тұруы мүмкін:

- 1) файлды құру және ашу;
- 2) нұсқағышты орнату;
- 3) файлды” ұзындығын анықтау;
- 4) керекті байтты, блокты оқу; байтты, блокты жазу; файл толтыру;
- 5) файлды жабу;
- 6) файлды жою.

Кейде барлық файлдарды табу керек болады (мысалы: TXT ке”еюі бар барлық файлдар не EXAMPLE аты бар файлдар). Ал бұл 4Eh қызметімен атқарылады.

; каталогтарды іздеуді бастау

|     |     |      |                             |
|-----|-----|------|-----------------------------|
| mov | AX, | 4Eh  |                             |
| lea | DX, | path |                             |
| mov | CX, | 10h  | ; барлық каталогтарды іздеу |
| int | 21h |      |                             |

4Fh-келесі файлды табу.

|       |     |      |      |                     |
|-------|-----|------|------|---------------------|
| next: | mov | AX,  | 4Fh  |                     |
|       | lea | DX,  | PATH |                     |
|       | mov | CX,  | 10h  |                     |
|       | int | 21h  |      |                     |
|       | cmp | AX,  | 18   | ; каталогты” со”ы ? |
|       | jne | next |      |                     |

Ол үшін 1Ah қызметі арқылы 43 байттан көп мөлшермен DTA (Disk transfer area) дерек алмасу аймағын ұйымдастыру керек немесе 2Fh қызметі арқылы DOS-пен құрылған дерек беруді” аумағыны” адресін алуға болады.

**1Ah функциясы** (DTA) дискімен алмасу аймағыны” адресін орнатады.

```
mov    AH,    1Ah
lea     DX,    DTA
int     21h
```

; деректер

```
DTA      db 43 dup (0)
```

**2Fh функциясы** (DTA) дискімен алмасу аймағыны” адресін анықтайды.

```
mov     AH,    2Fh
int     21h
mov     word ptr DTA, DX
mov     word ptr DTA+2, ES
```

; деректер

```
DTA      dd    0
```

DTA есебінде DOS PSP аумағын 80h байттан аяғына дейін қолдануы белгілі. DOS DTA-да табылған файл туралы ақпаратты орналастырады (атрибуттар, құрылу уақыты, дата, өлшемі т.т). Әсіресе DTA-ны” 1Eh...2Ah байттарында ASCIIZ қатарындағы файлды” аты мен ке”еюі сақталады. Файлды іздеген кезде 4Eh қызметі жандандырылады.

DS:DX регистрлеріне ASCIIZ қатарыны” адресі орналасады да, CX регистріне-ізделген файлды” атрибуттарыны” комбинациясыны” коды сақталады.

0-қалыпты файл , атрибутсыз

1-тек оқу үшін

2-жасырын

4-жүйелік

8-томны” белгісі

10h-каталог

20h-архивтендіру атрибуты

Егер іздеу атрибуттары айқын болса, қалыпты және атрибутты файлдар ізделеді.

Файл табылса CF=0 қызмет қайтарылады, ал ASCIIZ қатарындағы файлыны” аты мен ке”еюі DTA-ға 1Eh...2Ah байттарына енгізіледі. Файлды” атын алып, 3Dh қызметіні” көмегімен оны ашуға және одан әрі керекті операцияларды жүргізуге болады (оқу, жазу т.т).

Келесі файлдарды іздеу 4Fh қызметімен жүзеге асырылады, ол да 4Eh сияқты.

Керек болса 4Fh қызметін (CF=1 каталогта барлық файлдар біткенше) дүркіндете жандандыруға болады.

### **3.2. MS-DOS байланысыны” векторлық кестесі**

MS-DOS операциялық жүйесінде негізгі басқару блоктар байланысыны” кестесі бар.

Векторлық кестені” байланысыны” адресін құжатталмаған 52h int 21h функциясы көмегімен алуға болады. Бұл функция кесте адресін ES:BX регистріне қайтарады. BIOS мәліметтер аймағынан кейін MS DOS мәліметтер аймағы орналасқан. Мұнда MS DOS-ты” ішкі айнымалылар мен мәліметтер құрылымдары орналасқан.

Мәліметтерді” негізгі құрылымдары бұтақтар түрінде ұйымдастырылған. Байланысты” векторлық кестесі тамыры болып табылады, себебі басқа құрылымдарды” адрестерін сақтайды: MCB жадыны басқару блоктарыны” тізімі, MS-DOS құрылғыларыны” басқару блоктарыны” тізімі, файлдар кестесі, дискілік буферлер және т.б.

Байланысты” векторлық кестесінде басқа да керекті ақпарат болады, ол операциялық жүйе мәліметтеріні” барлық ішкі құрылымдарына жол ашып береді. Мысалы, операциялық жүйені” барлық резидентті және жүктелетін драйверлеріне қол жеткізу мүмкін. Жүйеде қандай дискілік құрылғылар орнатылғанын және оларды” қасиеттерін анықтауға болады. Операциялық жүйені” мәліметтер құрылымдарыны” пішімдерін біле отырып, программалық қамтаманы жөндеу кезінде қателік жағдайларды талдауға болады, құрылғыларды” конфигурациясын және жүйені” ішкі жағдайын көрсететін программаларды жасауға болады.

3.4-кестеде байланысты” векторлық кестесіні” негізгі өрістері келтірілген. Кестені” толық варианты В қосымшада берілген.

3.4-кесте

Байланысты” векторлық кестесіні” пішімі

| Ығысу | Байт саны | Өріс аты | Сипаттама                                                                                              |
|-------|-----------|----------|--------------------------------------------------------------------------------------------------------|
| -2    | 2         | mcb_seg  | Бірінші MCB-ді” сегменттік адресі                                                                      |
| 0     | 4         | dev_cb   | Диск параметрлер бірінші басқару блогына (MS-DOS Device Control Block) нұсқағыш                        |
| 4     | 4         | file_tab | Бірінші (SFT) жүйелік файлдар кестесіне нұсқағыш                                                       |
| 8     | 4         | clock_dr | CLOCK\$ белсенді құрылғыны” тақырыбына нұсқағыш                                                        |
| 12    | 4         | con_dr   | CON белсенді құрылғыны” тақырыбына нұсқағыш                                                            |
| 16    | 2         | max_btbl | Бөлек блоктардағы мәліметтерді жіберуге арналған құрылғылар үшін блокты” максимальды көлемі (байттағы) |
| 18    | 4         | dick_buf | Дискілік буферлерді сипаттайтын құрылымға нұсқағыш                                                     |
| 12    | 4         | drv_info | Құрылғылар туралы ақпарат массивіне нұсқағыш                                                           |
| 26    | 4         | fcv_tabl | FCB кестесіне нұсқағыш                                                                                 |
| 26    | 4         | fcv_size | FCB кестесіні” өлшемі                                                                                  |
| 32    | 1         | num_bdev | Орнатылған блокты құрылғыларды” саны                                                                   |
| 33    | 1         | lastdriv | Config.sys файлындағы LASTDRIVE мәні (келісім бойынша-5)                                               |
| 34    | 18        | null_dr  | NUL құрылғы драйверіні” тақырыбы. Бұл драйвер MS-DOS-тағы драйверлер тізімінде әрқашанда бірінші       |

МСВ бірінші блогыны” сегментіні” адресін сақтайтын - mcb\_seg өрісі. Осы мәнді біле отырып, жады блоктарын тексеруге және керек кезінде құрылымын өзгертуге болады. Мысалы, кейбір вирустар жады блогыны” со”ғы бос орындарыны”, өз денесін көшіре отырып, көлемін жасанды түрде азайтады. МСВ блоктарыны” құрылымын біле отырып, жадыдан керек емес резидентті программаларды өшіруге болады.

Dev\_cb өрісі, құрылғылар туралы анықтама ақпарат алу және төменгі сатыдағы дискіге қол жеткізу үшін дискілік құрылғыларымен басқарудағы блокты қолдануға мүмкіндік береді.

File\_tab өрісі, SFT файлдар бірінші жүйелік кестесіні” адресін сақтайды. Бұл кестеде әрбір ашық файл үшін әртүрлі ақпарат сақталады: файл үшін тағайындалған идентификаторлар саны (яғни қанша рет файл ашылған), файлды қолдану режимі, файлға үлестірілген дискіні” бірінші кластеріні” нөмірі және т.б.

Барлық бұл ақпарат кластерлер де”гейінде файлға қатынас құруын ұйымдастыру кезінде қажет болуы мүмкін, мысалы, файлдарды рұқсатсыз көшіруден сақтау жүйесінде.

Clock\_dr және con\_dr өрістерінің көмегімен, программа, сәйкесінше CLOCK\$ және CON консол драйверлеріне қатынас құрып алады. Бұлар программадан драйверлерді тікелей шақыруды ұйымдастыруда керек болады.

Мах\_btbl өрісі құрылғы блогының көлемінен тұрады және жеке блоктармен деректерді ауыстыруды атқарады. (Мысалға – диск драйвері). MS-DOS-тың 6.22 версиясында блоктың көлемі 512 байтқа тең.

Drv\_info өрісінде, дискілік құрылғы туралы мәлімет сақталатын, массивке нұсқағыш көрсетілген. Бұл массивта әрбір дискіге ағымды қатынас жолын, каталогқа үлестірілген дискінің бірінші кластерінің нөмірін және дискілік құрылғының сәйкес басқару блогының адресін табуға болады.

Осы массивті талдау арқылы, программа жүйеде орналасқан дискілік құрылғылар санын және олардың параметрлерін анықтайды.

Fcb\_tabl өрісінде FCB блоктар кестесіне нұсқағыш орналасқан. Осы кестенің көлемі fcb\_size өрісінде жазылған және config.sys файлында орналасқан fcbs=xx операторымен анықталады.

Lastdrv өрісі config.sys файлында орналасқан lastdrive операторы параметрінің мәнін сақтайды. Оны жүйедегі дискілік құрылғылардың максимальды санын анықтау үшін қолдануға болады.

Nul\_bdev өрісі дискілік құрылғылардың шынайы қолданылатын санын сақтайды.

Nul\_dr өрісі ешқандай керекті мәлімет сақтамайды. Тек оның адресі - бұл жерде тізбектегі ең бірінші MS-DOS драйвері орналасқан мәнін береді. Nul\_dr өрісінің адресін біле отырып, барлық тізбекті тексеруге және керекті драйверді іздеуді ұйымдастыруға болады.

## 4. ЖАДЫНЫ ЖӘНЕ ПРОЦЕСТЕРДІ БАСҚАРУ

### 4.1. Жадыны үлестіруіні” жүйелік құралдары

MS DOS жүйесі орындалатын программалар ретінде .COM және .EXE — екі типті файлдарды қолданады. .COM типті программа бірсегментті болады. Программалық қатарлар, деректерді” және стекті” аймақтары бір сегментте орналасады. Сегментті” максималды көлемі — 64 Кбайт. .COM типті файлдар дискіде жадыны” абсолютті бейнесі түрінде сақталады және жедел жадыға жүктеу процесінде олар өзгермей тасымалданады.

.EXE типті программаларды” кез-келген ұзындығы және программа, деректер және стектер сегменттерді” кез-келген саны болуы мүмкін.

.EXE файлда программадан басқа арнайы жүйелік аймағы бар. Оны тақырып дейді. Тақырыпта программаны” сипаттамалары (мысалы, программаны” ұзындығы, бақылау қосындысы, кіріс нүктені” адресі және т.б.) және жадыға жүктеу кезінде программалық кодаларын баптау туралы ақпарат сақталады.

Қазіргі уақытта қолданбалы программаларды” көбісі .EXE пішімінде жасалынады. .COM пішімі резидентті программаларды, драйверлерді және үзулер ө”деуіштерін жазу негізінде қолданылады.

Кез-келген программада PSP (программалық сегментті” префиксі) деген арнайы жүйелік кестесі болады. Оны” көлемі — 256 (100h) байт шамасында. .COM типті программаларда PSP-ға арналған орын ORG 100h операторымен беріледі. Программаны жадыға жүктеу кезінде EXEC функциясы PSP аймағын толтырады. EXE типті программаларда PSP-ға арналған орын қалдырылмайды, ол программаны жүктеу кезінде оған орналастырылады.

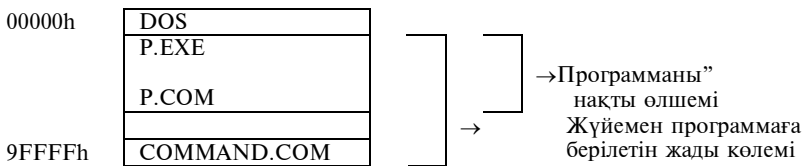
PSP-ны” мазмұны DOS-пен программаны орындау процесінде қолданылады. Қолданбалы программалар да PSP мазмұнына қатынаса алады. PSP мазмұны D қосымшасында көрсетілген.

Программаны жадыға жүктеу кезінде DOS оған барлық болған жадыны бөліп береді (4.1-сурет).

Жадыны” осындай берілу алгоритмі (EXE типті программаларға) жүктелетін модульдегі файлды” тақырыбында

0Ch ығысуы бар өрісіні” мәнімен анықталады. Келісім бойынша (по умолчанию) құрастырғыш бұл өрісті 0FFFFh мәнімен толтырады.

Дұрыс программа артық жадыны босатады. Бұл құрылғыларды” драйверлері және резидентті программалар үшін де өте ма”ызды. Егер драйвер жұмыс кезінде артық жадыны босатып алмаса, онда жүйе жұмыс істей алмайды, өйткені қолданбалы программаларды жүктеуге орын болмайды. Бірақ та егер программаға қосымша жады керек болса (мысалы, есептеу нәтижелерін орналастыру үшін), онда жадыны” керек блогын беру туралы DOS-тан сұрауға болады.

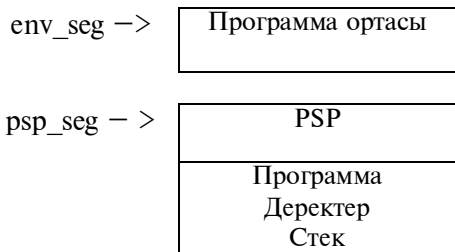


4.1-сурет. Жүктелетін программаға жадыны” берілуі

DOS бос жадыны” көлемі бойынша хабар береді.

Жадыға жүктелген программаны программалау үшін үш ма”ызды компоненті бар:

- программа ортасы
- программа сегментіні” префиксі
- программа өзі



4.2-сурет. Программаны” өз қоршау ортасымен жадыда орналасқаны.

env\_seg – программа ортасыны” сегменттік адресі; psp\_seg – программа сегментіні” префиксіні” сегменттік адресі (ID – программаны” идентификаторы)

Бастапқы жүктеу процесінде DOS бастапқы программа ортасын құрады. Бұл программа ортасында командалық процессор COMMAND.COM және жандандырылған

программалар жұмыс істей алады. Программа ортасы ретінде символдық қатар түрінде жазылған айнымалыларды” мәндері сақталатын жадыны” аймағын түсінеміз.

Программа ортасыны” пішімі 4.3-суретте берілген.

Программа ортасы керек параметрлерді программаларға жіберу үшін пайдаланылады. Параметрлер SET жүйелік команда көмегімен программа ортасына енгізіледі.

‘ 1\_айнымалы = 1\_мән’ , 0

‘ 2\_айнымалы = 2\_мән’ , 0

‘ . . .

‘ n\_айнымалы = n\_мән’ , 0, 0, 1, 0

‘диск: \ жол \ файл\_аты.ке”ейту’ , 0

4.3-сурет. Программа ортасыны” пішімі  
(барлық сандар бір байттан орын алады).

Жүйелік және қолданбалы программалар программа ортасыны” ағымдағы мазмұнын талдайды да, өзіне керек параметрлерді алады. Жүйені конфигурациялау кезегінде программа ортасыны” көлемі SET COMSPEC= команда көмегімен беріледі. Бұл команда командалық процессорды” кейбір сипаттамаларын және орналасу орнын орнатады.

Мысалы:

SET COMSPEC=C: \DOS\_62\COMMAND.COM /P /E:400

Команда командалық процессор DOS\_62 каталогта орналасқанын, ол резидентті (/P кілт) болатынын және 400 байт көлемі (/E кілт) бар программа ортасында жұмыс істейтіні туралы жүйеге хабар береді. Егер программа ортасына айнымаларды” көп санын енгізу керек болса, онда берілген команда бастапқы программа ортасыны” көлемін көбейтуге мүмкіндік береді.

Ортаға жүйемен енгізілген программасы бар файлды” толық спецификациясы жадыда белгісіз программаны” ортасын тапқанда осы программаны” атын білуге мүмкіндік береді (программада және файлды” жүктелетін модулінде оны” аты туралы ақпарат жоқ).

Әдетте DOS ортаны программаны” алдында орналастырады. Ортаны” сегменттік адресі жүйемен программаны” PSP-сіне жазылады.

Орта параграфты” шекарасында басталады және оны” тұрған орны сегменттік адреспен сипатталады.

DOS жедел жадыны кез-келген ұзындығы бар блоктармен бөліп береді. Блок көлемі параграфпен беріледі және оны” мәні 0...FFFFh шамасында.

Жадыны басқару блоктар көмегімен, бос және бос емес жадыны есепке алуға болады. MCB (Memory Control Block)-16-ты байттық құрылым. Әрбір MCB жадыда өз блогыны” алдында орналасады.

MCB пішімі 4.4-суретте көрсетілген.

Жадыны динамикалық басқару үшін DOS-ты” келесі функциялары қолданылады:

48h - жады блогын бөліп беру (7, 8)

49h - жады блогын босату (7, 9)

4Ah - берілген жады блогыны” көлемін өзгерту (7, 8, 9).

Байттар:

|           |                   |   |                |   |        |   |   |                                                  |   |    |    |    |    |    |    |
|-----------|-------------------|---|----------------|---|--------|---|---|--------------------------------------------------|---|----|----|----|----|----|----|
| 0         | 1                 | 2 | 3              | 4 | 5      | 6 | 7 | 8                                                | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Тү-<br>рі | Иесіні”<br>адресі |   | Блок<br>көлемі |   | Резерв |   |   | Блок – программа болса, онда<br>программаны” аты |   |    |    |    |    |    |    |

Типті” мәні: ‘M’ = 4Dh – арасындағы блок

‘Z’ = 5Ah - со”ғы блок

4.4-сурет. MCB жадыны басқару блогыны” пішімі

Жадыны үлестіру int 21h-ты” үш функциясы жады блогы басыны” 16-шы адресін қолданады. Бұл адрес блок басталатын сегментіне сәйкес келеді. Үш функция үшін BX-та параграф саны болады, яғни блок көлемі. Егер функция орындалмаса, онда тасымал жалауына “1” жазылады, ал AX-ке қате кодасы қайтарылады.

Қателерді” үш кодасы мүмкін болады:

7 - жады басқару блогы бұзылған

8 - функцияны орындау үшін жады жетіспейді

9 - жады блогыны” адресі дұрыс емес.

Мысал.

; 1024 байт көлемі бар блокты бөліп беру

mov AH, 48h ; жады блогын беру функциясы

mov BX, 64 ; 64 параграф керек

int 21h

jc error ; операция сәтті орындалмаса, қатені

; ө”деу

mov BLOCK\_SEG, AX ; сәтті болса – блок адресін сақтаймыз

; блогымызды босатамыз

```
mov AX, BLOCK_SEG ; блокты” бастапқы адресін алу
mov ES, AX          ; оны ES-ке жазу
mov AH, 49h         ; жады блогын босату функциясы
int 21h
```

Бұл мысалда әуелі 1024 байт бар блогы беріледі. BX-параграфпен берілген блок көлемі  $(1024/16=64)$ .

Мысалды” екінші бөлігі осы блокты босатады (оны программаны аяқтау кезінде талап етеді).

Осы жағдайда AX-қа қайтарылған мән ES-ке жазылады. DOS блок көлемін бақылап, қанша параграф санын босату керегін біледі.

Енді 4Ah функциясын қолдану мысалын келтіреміз. ES-та PSP сегментіні” мәні бар. Бұл мән программа жүктелген жадыны” е” бірінші байтыны” адресін көрсетеді. Бұл мән программаны іске қосқанда ES-ке жазылады.

SETBLOCK қолдану үшін бұл функцияны программаны” басына шақыру керек (ES өзгеру алдында) немесе ES-ты” бастапқы мәнін сақтау керек. BX-ке параграф бойынша блокты” керек көлемін жазу керек. Бұл көлемді анықтау үшін программаны” со”ына қосымша сегментті қою керек. Программа осы сегментті” тұрған орнын анықтап және оны бастапқы сегментімен салыстырып, программаға керек жадыны” көлемін алады. Программаны жүктеу кезінде ES-та және DS-та PSP-ды” е” басыны” параграф нөмірі болады; .COM файлдар үшін PSP CS, DS, ES арқылы беріледі.

; жадыны босату

```
mov BX, ZSEG ;(программа со”ы+1)–ды” параграф нөмірі
mov AX, ES   ; программа басыны” параграф нөмірі
sub BX, AX   ; параграф бойынша программаны” көлемі
mov AH, 4Ah  ; функция нөмірі
int 21h      ; жадыны босатамыз
jc memory_error ; қатені тексереміз
```

; бос сегмент

```
ZSEG segment 'abcd'
ZSEG ends
```

Жалған сегментімізге кез-келген класты беру керек, өйткені транслятор және программа құрастырғыш (компилятор) сегмент кластарын бақылайды.

Бір сегментті .COM программаны” көлемі басқаша анықталады. Ол программаны” со”ы мен басы ағымдағы адрес мәндеріні” айырмасы мен PSP көлеміне те”. 4Ah функцияны қолдану кезінде, .COM программаны жүктеу кезінде, 64 Кбайт сегментіні” бос ке”істігі стекке беріледі. SP - стек көрсеткішіне 0FFFFh сан жазылады. Ол стек со”ын білдіреді. Стэк көлемін қажет шамасына келтіру үшін, артық жадыны босату алдында стек көрсеткішін, стекке арнайы берілген аймағыны” со”ына қою керек:

```

text    segment 'code'
        org 100h
main     proc
        . . .
        mov SP, offset newstk
main     endp
        . . .           ; деректер
        dw 64          dup (?) ; стек аймағы
newstk=$
text     ends

```

Параграф бойынша программа көлемі  
 $(newstk - main + 100h + 0Fh)/16$

өрнек мәніне те” болады. Бұл формулада 100h - PSP өлшемі, 0Fh - келесі параграфқа дейін до”гелектеу. Барлық қосындыны 16-ға бөлу параграф санын береді.

COM-программаны қарастырайық. Осы программа орындалу кезінде программамен жұмыс істейтін кейбір буферді” көлемі есептеледі және талап етілген жады динамикалық түрде программаға беріледі.

Программада келесі үзінділер болуы керек:

- параграф бойынша керек жадыны” көлемін анықтау
- стек көрсеткішін жа”а орынға тасымалдау
- артық жадыны босату
- бір немесе бірнеше сегмент түрінде жадыны” керек көлемін

бөліп беру

- программаны” талабына қарай бөліп берген сегменттерді қолдану (файлды оқу үшін, деректерді сақтау үшін )
- бөліп берілген жадыны босату
- программаны аяқтау

Программаны” құрылымы келесі түрде болуы мүмкін:

```
text    segment    ‘code’
        org 100h
main    proc
; программаны” е” со”ына жа”а орынға стекті орналастыру
        mov     SP,    offset newstk
; артық жадыны босату (стек со”ынан жедел жадыны” со”ына дейін)
        mov     AH, 4Ah    ; жады блогыны” көлемін өзгерту функциясы
        mov     BX, (newstk-main+10Fh)/16 ; программаны” нақты көлемі
        int     21h
; динамикалық түрде программаға бөліп беретін жады аймағын
; қандай да бір әдіспен анықтау. Бұл шама (байт бойынша) AX-та
; болу керек
        add     AX, 0Fh    ; параграф санын дө”гелектеу үшін
        mov     DX, 0      ; бөлінгішті” үлкен жартысы
        mov     BX, 16     ; бөлгіш (параграф өлшемі)
        div     BX ; AX-та бөлінді, DX-та қалдық бізге керегі жоқ
; талап етілген жадыны динамикалық түрде бөліп береміз
        mov     BX, AX     ; талап етілген параграф саны
        mov     AH, 48h    ; жадыны бөліп беру функциясы
        int     21h
        mov     ES, AX ; бөліп берген сегментке ES-ті баптаймыз
; бөліп берген блогымен жұмыс істейміз
; талап етілген ығысу DI регистрде болса, оған жазу мысалы
        mov     word ptr ES:[DI], 1234h
; талап етілген ығысу SI регистрде болса, одан оқу мысалы
        mov     AX, word ptr ES:[SI]
; бөліп берген блогты босатамыз. ES бұл блокты” адресін көрсетіп
; түр
        mov     AH, 49h    ; жадыны босату функциясы
        int     21h
; программаны аяқтаймыз
        . . .
main    endp
; деректер
        . . .
dw      128    dup (0) ; стек аймағы
newstk=$
text    ends
```

## 4.2. Туынды процестерді” ұйымдастырылуы

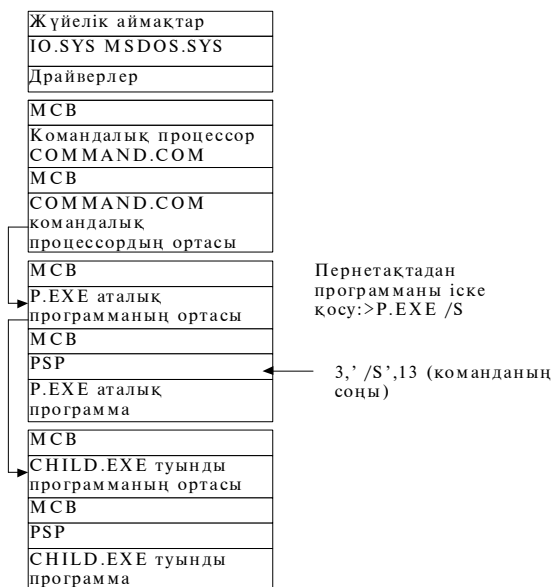
Программалар орындалу үшін Exec (Int 21h, 4Bh функциясы) DOS функция көмегімен жадыға жүктелінеді.

Жадыға жүктелген программа жадыны” 4 блогын алады. Олар: программаны” ортасы (окружение) оны” MCB жадыны басқару блогымен бірге және программаны” өзі оны” MCB-мен бірге (4.5- сурет).

DOS-ты” бастапқы жүктеу процесінде командалық процессорды” ортасы құрылады. Бұл ортада COMSPEC, PROMT және PATH айнымалылары бар. Олар AUTOEXEC.BAT файлдан ортаға жазылады.

Мысалы: COMSPEC=C:\DOS\COMMAND.COM  
PROMPT= \$P\$G  
PATH=C:\; C:\DOS; C:\TOOLS

DOS- ты бастапқы жүктеген со”, белсенді программа ретінде командалық процессор болады. Бірінші қатарда COMMAND.COM командалық процессорды” файлыны” жолы көрсетілген. Жүйелік немесе қолданбалы программалармен өшіріліп қалған командалық процессорды” транзитті бөлігін жүктеуге қажет болғанда қолданады.



4.5-сурет. Туынды процесті іске қосу

PROMPT айнымалысы бар қатар жүйелік шақыруды” түрін анықтайды (ағымдағы каталогқа толық жол және стрелканы” бейнесі).

PATH айнымалысы бар қатар пайдаланушымен іске қосылған программаны дискіде іздегенде, DOS автоматты түрде қарап шығатын жолдарды анықтайды.

SET командасы көмегімен пайдаланушы қоршау ортасына өз айнымалыларын енгізе алады. DOS-ты бастапқы жүктегеннен кейін белсенді программа (ағымдағы процесс) ретінде командалық процессор болады. Мысалы, P.EXE программаны іске қосқанда COMMAND.COM EXEC функция көмегімен осы программаны жандандырады және оған өз ортасын жібереді. Ортаны” қатарлары жадыны” жа”а аймағына жүктелетін программаны” алдына көшіріледі. Егер жүктелген программа туынды процесті (келесі де”гейді”) жандандырса, онда ол оған өз ортасын жібереді. Аталық процесс ортаны өзгермейтін түрде немесе ортаға жа”а ақпаратты енгізіп жібере алады. Программа аяқталғаннан кейін басқару аталық процеске қайтарылғанда,

туынды программаны” ортасы жойылады. Орта блогыны” максималды өлшемі 32Кбайт шамасында болады.

Программаны жандандырғанда EXEC функциямен қалыптастыратын деректерді” екінші құрылымы ретінде PSP болады. PSP (өлшемі 256 байт) жүктелетін программаны” құрамына енгізіледі. DOS 2Ch адрес бойынша программа ортасыны” сегменттік адресін, ал 80h - адресі бойынша – командалық қатарды” параметрлерін (команданы” со”ын /s) орналастырады. Адресі 80h байтта команданы” параметрлеріні” ұзындығы (13 кодасы оған кірмейді) болады. Адресі 81h-тан бастап параметрлерді” өзі орналасады.

Егер қолданбалы программаға іске қосу кезінде қандай болса да параметрлері берілсе, онда программада PSP-дан команда со”ын алатын және оны талдайтын қатарлар болу керек.

Туынды программаны жүктеу және орындау үшін аталық программа Int 21h-ты” 4Bh функциясын қолданады.

Функцияны шақыру кезінде: AH=4Bh

AL - ішкі функцияны” коды (0,1,2,3,5)

DS:DX - туынды программаға жолды көрсететін ASCIIZ-қатар

ES:BX - параметрлер блогыны” адресі

4Bh функциясы қатесіз орындалса CF=0 болады, қате болса CF=1 және AX регистрде қате коды көрсетіледі.

Қате қатар:

1 – ішкі функцияны” коды дұрыс емес

2 – іске қосылатын программаны” файлы табылған жоқ

3 – көрсетілген жол табылған жоқ

4 – ашық файлдарды” саны өте көп

5 – қатынауға болмайды

8 – программаны жүктеу үшін жады жетіспейді

10 – орта блогыны” ұзындығы 32 Кбайттан артық

11 – іске қосатын .exe файлды” форматы дұрыс емес

Ішкі функциялар (AL):

0 – программаны жүктеу және орындау

1 – программаны жүктеу және орындау (құжатталмаған ішкі функция)

3 – программаны оверлей ретінде жүктеу (PSP блогын құрып алмай)

5 – программаны орындау үшін дайындау

Параметрлер блогы Exec функцияға керек ақпаратты беру үшін қолданылады. Оны” ішінде келесі объектілерді”:

- ортаны” (сегменттік адрес, сөз)
- командалық қатарды” параметрлерді” (толық адрес, 2 сөз)
- файлдарды басқару екі блогыны” (FCB) (толық адресстер, 2 қос сөз) адресстері бар.

Мысалы, параметрлер блогы және онымен байланысқан деректер құрылымдары келесі түрде болуы мүмкін:

```
Parmblk dw  envirseg      ; ортаны” сегменттік адресі
         dd  cmdtail      ; команда со”ыны” адресі
         dd  fcb1         ; бірінші FCB-ді” адресі
         dd  fcb2         ; екінші FCB-ді” адресі
Chname  db  ‘CHILD.EXE’, 0 ; туынды процесі” аты
Cmdtail db  9, ‘FILE.TXT’, 0Dh ; команданы” со”ы
fcb1    db  37 DUP (0)      ; бірінші FCB
fcb2    db  37 DUP (0)      ; екінші FCB
envirseg segment para ‘envir’ ; ортаны” сегменті
         db  ‘COMSPEC=D:\DOS\COMMAND.COM’, 0
         db  ‘WORKFILE=A:\FILES’, 0
         db  0
envirseg ends
```

Ортаны” адресі тек қана сегменттік адресстен тұрады, өйткені орта блогы әрқашан сегментті” шекарасынан басталады. Егер аталық процесс өз ортасын туынды процеске берсе, онда арнайы сегмент құрылмайды. Сондықтан, параметрлер блогында ортаны” адресі ретінде 0 беріледі.

Команданы” со”ы туынды программаға шақыру параметрлерін (мысалы, жұмыс файлдарыны” аттарын) және кілттерді беру үшін қолданылады. Кілттер программаларды” жұмыс режимдерін анықтайды. Туынды программаны” жүктеу процесінде DOS команданы” со”ын параметрлер блогынан туынды программаны” PSP-сіне жібереді.

Егер туынды программаны шақыру кезінде оған параметрлер берілмесе, онда команданы” со”ы келесі түрде болады:

```
db      0, 0Dh
```

немесе параметрлер блокта команданы” со”ыны” адресі орнына 0 беруіне болады.

(FCB) файлдарды басқару блоктары қазіргі уақытта өте аз қолданылады. Егер жүктелетін программа FCB-ді қолданбаса,

онда оларға орын берілмейді, ал FCB-ды” адрестері үшін өрістеріне 0 жазылады.

Е” қарапайым вариантта (аталық программаны” ортасы жіберіледі, ал команданы” со”ы мен FCB болмаса) EXEC функциясын шақыру үшін деректерді” құрылымы келесі түрде болады:

```
parmblk    dw    7 DUP (0)
chnamedb   'CHILD.EXE', 0.
```

Туынды процесті жандандыру (активизация) алдында аталық программа туынды программаны жүктеу үшін жеткілікті жадыны” көлемін босатуға міндетті.

Туынды программа аталық программаны” барлық ашық дескрипторларына (файлдар мен құрылғыларына) ие болады. Екеуі де (программалар) файлдарды бірге қолдана алады. Осы кезде туынды программада орындалып жатқан енгізу-шығару операциялары аталық программадағы дескрипторлар күйіне әсер етеді.

Туынды процесті жандандырудан кейін DOS аталық программаны” орындалуын, туынды программаны” жұмысы біткенше тоқтата тұрады.

Туынды процесс аталық процеске қайтару (біту) кодын бере алады. Коданы” мәндері 0-ден 255 дейін болуы мүмкін. 0 – дұрыс аяқталуыны” кодасы, басқаларына қате туралы хабар береді. Қайтару кодасы 4Ch функция көмегімен беріледі:

```
mov    AH,    4Ch    ; процесті” аяқталу функциясы
mov    AL,    errcode ; аяқталу коды
int21h                ; аталық процеске басқаруды беру
Аяқталу кодасын алу үшін аталық программа 4Dh
```

функциясын орындау керек:

; Аталық программа

; деректер өрістері

```
parmblk dw    7 DUP (0)
```

```
chname  db 'B:\info_01.EXE',0 ; туынды программаға жол
```

; программалық жолдар

; программаны” шегінен бастап жадыны босату

...

; Туынды программаны” шақыруын дайындау

```
mov    AH,    4Bh    ; Exec функциясы
```

```
mov    AL,    0      ; туынды программаны жандандыру
```

```

; режімі (тәртібі)
mov BX, seg parmbk ; ES регистрі параметрлер блогыны”
mov ES, BX ; сегменттік адресіне қалыптастыру
mov BX, offset parmbk ; BX=параметрлер блогыны” ығысу
; адресі
mov DX, offset cname ; DX=туынды программаны” атыны”
; ығысу адресі
int 21h ; туынды процесті жандандыру
; аяқталған туынды программадан аяқталу кодын аламыз
mov AH, 4Dh ; аяқталу кодасын алу функциясы
int 21h ; аяқталу кода үшін DOS-қа қатынасу
; алынған аяқталу кодасын талдау
cmp AL, 0 ; 0 мәні қайтарылды ма?
je norm ; ия, қалыпты аяқталу
cmp AL, 1 ; мәні 1?
je err1 ; ия, кодасы 1 қатені ө”деуге көшу
cmp AL, 2 ; мәні 2?
je err2 ; ия, кодасы 2 қатені ө”деуге көшу
. . .
; туынды программа
; деректер өрістері
errcode db 0 ; аяқталу кодасына арналған байт
; программалық жолдар
. . .
; программа орындалу кезінде кодасы 1 қате пайда болды
or errcode, 1 ; қайтару кодасында 0-ші битке “1”
; орнатамыз
; программа орындалу кезінде кодасы 2 қате пайда болды
or errcode, 2 ; қайтару кодасында 1-ші битке “1”
; орнатамыз
; туынды программаны аяқтау
mov AH, 4Ch
mov AL, errcode
int 21h

```

#### 4.2.1. Туынды программаға ақпаратты беру

Аталық программада өз деректер аймақтарыны” адреестерін туынды программаға беру үшін жадыны” екі аймағы бар:

– үзулерді” бос векторлары

– есепаралық байланыстар аймағы

60h...66h векторлар пайдаланушыларды” үзулеріне арналған.

Векторларды деректерді” адреестерін беру үшін қолдануға болады.

Туынды программаға ақпаратты беру үшін пайдаланушыны” векторларын қолдану әдістемесі:

; аталық программа

; деректер сегментіндегі деректер өрісі

array1 dw 1024 dup (0)

; программалық жолдар

mov AH, 25h

; үзу векторын толтыру функциясы

mov AL, 60h

; пайдаланушы векторы

mov DX, offset array1

; DS:DX-дегі массивті” толық адресі

int 21h

; үзу векторына жіберіледі

; DOS-ты” EXEC 4Bh функция көмегімен туынды программаны шақыру

...

; туынды программа

mov AH, 35h

; үзу векторын алу функциясы

mov AL, 60h

; пайдаланушы векторыны” нөмірі

int 21h

; DOS ES:BX регистрлерінде векторды” ішкі мазмұнын қайтарды

; туынды программа аталық программаны” деректер өрісіне тікелей

; қатынай алады

mov AX, ES:[BX]

; array1 массивіні” 1-ші сөзін оқу

mov DX, ES:[BX+2]

; array1 массивіні” 2-ші сөзін оқу

Екінші тәсілде программа аралық ақпарат алмасуда, есепаралық байланыстар аймағын пайдаланады. Ол BIOS деректер аумағыны” е” со”ында орналасады 40h:FOh...40h:FFh (4 қос сөз). Бұл аймақты жүйе пайдаланбайды.

Есепаралық байланыстар аймағын пайдалану:

; аталық программа

; деректер өрісі деректер сегментінде

array2 db 10000 dup (0)

; программалық жолдар

```
mov AX, 40h      ; ES регистрін BIOS-деректер аумағыны”  
mov ES, AX       ; басына баптаймыз (настроим)  
mov word prt ES:OF0h, offset array2 ; array2 массивіні” ығысуы  
mov word prt ES:OF2h, seg array2 ; array2 массивіні” сегменті  
; DOS- ты” EXEC функция көмегімен программаны шақыру  
.  
.  
.  
; туынды программа  
mov     AX,      40h ; ES регистрін BIOS-деректер аймағыны”  
mov     ES,      AX   ; басына баптаймыз  
LES     BX,      dword prt ES:OF0h  
; туынды программа аталық программаны” деректер өрісіне тікелей  
; қатынай алады  
mov     AL,      ES:[BX] ; array2 массивіні” 1-ші байтын оқуы  
mov     AH,      ES:[BX+1] ; array2 массивіні” 2-ші байтын оқуы
```

#### 4.2.2. Ағымдағы программаны үзбей DOS командасын шақыру

Ол үшін аталық программадан Command.com командалық процессорды” екінші көшірмесі шақырылады. DOS командасын шақыру мүмкіндігі (мысалы, файлдарды” көшірмесін алу, дискеттерді тексеру т.б.с.с.) қазіргі кезде барлық қолданбалы программаларда қарастырылған (қосу жинақтау (интегрированный) программалық орталарында, электрондық кестелерде, деректер базасында, мәтінді процессорларда және т.б.с.с.).

Командалық процессорды Exec функциясы көмегімен шақыру үшін, оны” файлыны” толық (спецификациясын) жолын көрсету керек. Ол ағымдағы процесті” ортасынан алынады немесе программаны” деректер өрісінде тікелей берілуі мүмкін.

Командалық процессорға программа атын берген кезде (командалық жолды” параметрі есебінде) ол /C кілтінен басталуы керек (осындай мақсатқа қызмет ететін COMMAND DOS командасын шақырудағы сияқты). Бұл жағдайда команданы” со”ы мынандай түрде болуы мүмкін:

```
cmdtail db 10,' /C DIR A:', 0Dh.
```

Командалық процессорды” екінші көшірмесін жандандыруды” екі вариантын бөліп қарауға болады:

1. Командалық процессор /C кілтімен шақырылады және онда нақты программа көрсетіледі (DOS командасы). Басқаруды

алған со” командалық процессор сұратылған программаны жандандырады. Ол жұмыс істеп болған со”, басқару бірден аталық процеске беріледі. Бұл тәсілмен DOS командаларын ғана емес және сонымен бірге командалық файлдарды (Exec функциясыны” көмегімен тікелей жандандыруға болмайтын) да іске қосуға болады.

2. Командалық процессор кілтсіз және аты жоқ нақты программаны шақыра алады. Бұл жағдайда командалық процессор басқаруды алғаннан кейін DOS командасын пернетақтадан енгізуді күтеді. Пайдаланушы DOS-та қанша жұмыс істегісі келсе сонша істейді (яғни DOS кез-келген командасын немесе қолданбалы программаларды шақыра отырып).

Аталық процеске қайту үшін Exit командасын енгізу қажет.

## **5. ҰЗУЛЕРДІ ӨҢДЕУ**

### **5.1. Ұзулер контроллері**

Сыртқы аппараттық ұзулерді” сигналдары процессорға (intel 8259A) ұзулер контроллері арқылы кіреді.

IBM PC/XT-ны” аппараттық ұзулеріні” ұйымдастыруын қарастырайық (5.1-сурет).



5.1-сурет. IBM PC/XT машиналарында аппараттық үзулерінің ұйымдастырылуы

IRQ (interrupt request) - үзуге сұратуы.

Үзулер контроллерінің негізгі мақсаты – микропроцессорды” жалғыз кірісіне 8 құрылғыдан келетін үзулер сұратуларының” сигналдарын жіберу.

Int сигналы микропроцессорда үзу процедурасын (шақырады) бастайды. Int сигналдан басқа контроллер үзулерді өтейтін программасы шақырылатын векторының нөмірін МП-ға алып береді.

Жіберілетін вектордың нөмірі, негізгі векторы және үзуге сұрату келетін кіру жолының нөмірімен қосу арқылы құрастырылады.

Компьютердің бастапқы жәктеу кезінде негізгі вектордың нөмірі контроллерге автоматты түрде енгізіледі.

20h, 21h порттары арқылы контроллер программаланады. Аппараттық үзулердің вектор нөмірлері 8h, ..., Fh диапазонында жатады.

PC/AT машиналарда 2 контроллер қолданылады. Қосылатын құрылғылар саны 15-ке дейін болады (7-жүргізуші контроллерде және 8-бағынушы контроллерде).



5.2-сурет. IBM PC/AT машиналарында аппараттық үзулерінің ұйымдастырылуы

5.1-кесте  
(IBM PC/AT) үзулер векторлары мен компьютерді” құрылғыларыны” сәйкестігі

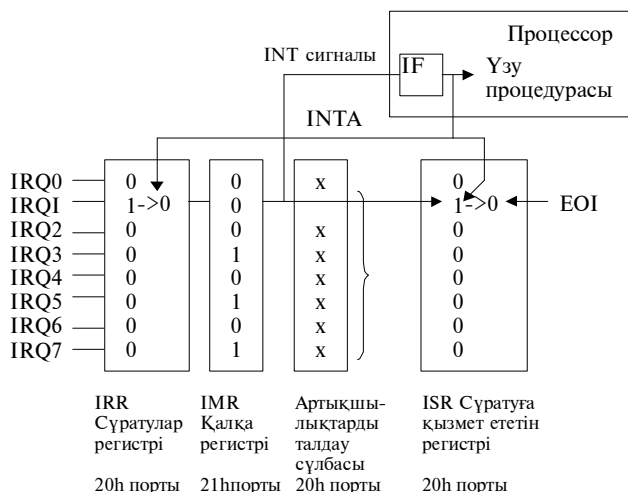
| Үзу деңгейі | Үзу векторы | Құрылғы                     |
|-------------|-------------|-----------------------------|
| IRQ0        | 08h         | Таймер                      |
| IRQ1        | 09h         | Пернетақта                  |
| IRQ2        | 0Ah         | Бағынушыдан кірісі          |
| IRQ8        | 70h         | RTC нақты уақытты сағаты    |
| IRQ9        | 71h         | Int 0Ah - қа бағытталған    |
| IRQ10       | 72h         | Резервтелген                |
| IRQ11       | 73h         | Резервтелген                |
| IRQ12       | 74h         | (PS/2) Мыс                  |
| IRQ13       | 75h         | Сопроцессор                 |
| IRQ14       | 76h         | Қатты диск контроллері      |
| IRQ15       | 77h         | CD-ROM                      |
| IRQ3        | 0Bh         | COM2 тізбекті порт          |
| IRQ4        | 0Ch         | COM1 тізбекті порт          |
| IRQ5        | 0Dh         | LPT1 параллель порт         |
| IRQ6        | 0Eh         | Йілгіш диск                 |
| IRQ7        | 0Fh         | LPT1 параллель порт принтер |

Контроллерді” ішкі құрылымын қарастырайық.

Ол 4 негізгі бөлшектен тұрады: енгізу сұратулар регистрі, қалқа (маска) регистрі, артықшылықтарды (приоритеттерді) талдау сұлбасы және сұратуға қызмет ететін регистр (5.3-сурет).

Барлық контроллерді” бөлшектері 8-биттік, әр енгізу сигналына —1 биттен келеді.

IRQ1-ге пернетақтадан сигнал сұрату регистріні” кірісіне келеді және регистрді” сәйкес битін 1-ге орнатады.



5.3-сурет. Үзу контроллердің логикалық құрылымы

Содан кейін сигнал қалқа регистріне келеді (21h порт арқылы программаланады). Қалқа битінде мәні 0 болса, сигнал өтуге рұқсат беріледі, ал мәні 1 болса – рұқсат берілмейді.

Сигнал қалқа регистрден өтіп, артықшылықтарды талдау сұлбасына келеді. Ол анықталған командаларды жіберумен 20h порт арқылы программаланады.

Артықшылықтарды талдау сұлбасын стандартты баптау кезінде IRQ сигналдарды” артықшылықтары сигналды” нөмірі осу ретімен төмендейді, яғни е” жоғары артықшылық IRQ0 сигналда, ал е” төмен артықшылық IRQ7 сигналда болады.

Үзуді сұрату сигналы, артықшылықтарды талдау сұлбасына өтіп:

1) сұратуларға қызмет ететін регистріні” кірісіне келеді және оны” битін 1-ге орнатуға рұқсат береді (бірақ орнатпайды);

2) сол уақытта сигнал МП-ды” INT кірісіне келеді. Егер IF үзуге рұқсат жалауы “1”-ге те” болса, онда МП INT сигналыны” келуін тіркейді. Ал егер IF=0 болса (cli команда арқылы), онда барлық аппараттық үзулерге тыйым салынады.

МП INT сигналды қабылдап INTA сигналды жібереді, ол контроллерге келіп екі әрекет жасайды:

1) сұратуларға қызмет ететін регистрді" битін "1" орнатады;

2) сұратулар регистрді" битін "0" орнатады. Сонымен, келген сұратуға МП қызмет ету процедурасын бастап, сол сұрату қызмет алатын топтарына өтеді.

Осы моменттен бастап сол ұзу контроллерді" кірісіне, құрылғыдан ұзу сигналы келуі мүмкін. Келген сигнал сұратулар регистрінде сақталады және контроллермен МП-дан қызмет етуін күтеді.

МП INTA сигналын ұзу контроллеріне жіберумен бірге жалаулар регистрінде IF разрядына "0" орнатып, барлық аппараттық ұзулерге тыйым салады (sti команда келгенше немесе кез-келген басқа тәсілмен IF разрядына "1" орнатқанша).

Сұратуларға қызмет ететін регистрді" разрядында, бірлік артықшылықтарды талдау сұлбасына әсер етеді. ISR-да "1"-ге орнатылған бит артықшылықтарды талдау сұлбасында ұзулерді" барлық де"гейіне (ағымдағы де"гейден бастап төмен қарай) тыйым салады.

ISR-ды" разрядын "0"-ге орнату үшін:

1) жүргізуші контроллер үшін 20h портына 20h кодасын жіберу керек;

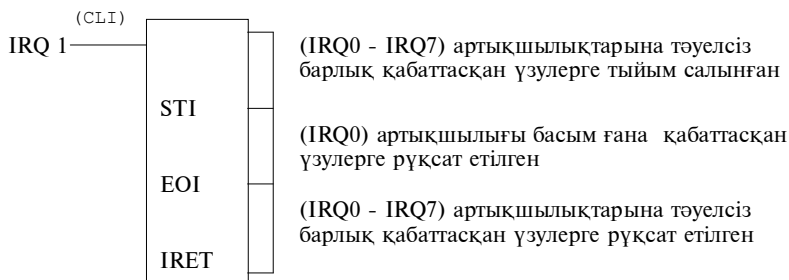
2) ал бағынушы контроллер үшін – A0h портына 20h кодасын жіберу керек.

20h кодасы - үзуді" со"ы, EOI – End Of Interrupt деп аталады. Үзуді" со"ы - 20h командасы кез-келген ұзулерді" ө"деуішінде болу керек.

Ұзулерді ө"деу программаларында (5.4-суретте) көрсетілген үш бөлімін белгілеуге болады:

1) МП, Int сигналын қабылдап, IF разрядына "0" орнатады, сонымен барлық ұзулерге тыйым салынады.

2) Программада орындалған STI командасы IF-ке "1" орнатады және ұзулер сұратуларға МП-ға өтуге рұқсат береді. Бірақ та, ағымдағы де"гейден бастап, контроллерде ұзулерді" барлық де"гейлері бұғатталған (заблокированные) түрде қалады. Ұзу ө"деуішті" жұмысын жоғары артықшылығы бар (IRQ0) ұзу сұратумен үзуге болады. Осындай оқиға қабаттасқан (вложенные) ұзулер деп аталады.



5.4-сурет. Үзулерді өңдейтін программаның жалпыланған құрылымы

3) EOІ командасы контроллердегі бұғаттауын алады, енді кез-келген де”гейіні” үзу сұратуы ө”деушіні” жұмысын үзе алады. Бірақ келесі оқиға болуы мүмкін: үзу ө”деушіні” жұмысы бірде”гейлі үзу сұратулар сигналымен үзіледі, яғни программаға қайта кіру пайда болады, сонымен жүйені” жұмысы бұзылады.

## 5.2. Үзулерді” қолданбалы және жүйелік ө”деуіштеріні” өзара әрекеттестігі

Үзу ө”деуішті” құрылымы және оны” программалық кешеніні” компоненттерімен әрекеттестігі келесі факторлармен анықталады:

- ө”деуішті инициализациялайтын үзулер аппараттық (сыртқы құрылғыдан) немесе программалық (INT командасы) болуы мүмкін;

- ө”деуіш программасы резидентті немесе транзитті болуы мүмкін;

- ө”дейтін үзуді” векторы бос немесе жүйелі болуы мүмкін.

Егер жүйелі вектор қолданылса, яғни үзуді жүйелі ө”деуішті” берілген түрі DOS құрамында бар болса, онда жүйелі ө”деуіш жа”а ө”деуіште өзгермейді оған тіркелінеді (сцепляется).

Жүйелі ө”деуішпен тіркелуі кезінде жа”а ө”деуіш өз функцияларын оны” алдында немесе содан кейін орындайды.

Үзу нәтижесінде қолданбалы ө”деуіш басқаруды алу үшін, оны” адресін тиісті үзу векторына қоюы керек. Ол үшін DOS-ты” INT 21h-ты” 25h функциясын қолдану керек. AL регистріне өзгерілетін вектор нөмірі, ал DS:DX регистріне – жа”а ө”деуішті”

адресі қойылады. Векторды” ескі мәндерін, берілген ұяшықтарда сақтау керек және программаны аяқтау алдында өз қалпына келтіру керек.

Негізгі векторды” мәнін алу үшін DOS-ты” 35h функциясы қолданылады. Оны шақыру үшін:

АН регистріне - 35h, AL-ға вектор нөмірі жүктелінеді. INT 21h орындалғаннан кейін ES:BX регистріне векторды” екі сөзді мәні жазылады.

Жүйе пайдаланбайтын үзулер ө”деуіші бар программаларды” құрылымы келесі түрде болуы мүмкін:

```

text      segment      'code'
intr      proc
...
        iret
intr      endp
main      proc
        push  CS          ; программалық сегментке
        pop   DS          ; адрестеуді орнату
mov       DX,  offset intr ; DS:DX -> intr
mov       AH,  25h        ; векторды орнату функциясы
mov       AL,  0          ; вектор нөмірі
int       21h
; үзу векторы инициализацияланған
; өрі қарай негізгі программаны” мәтіні
...
main      endp

```

Егер пайдаланушыға үзу ө”деуішті” жүйелік алгоритмдері жарамаса, ол тиісті векторды” мәнін өз ө”деуішіні” адресіне ауыстырып, үзулерді ө”деуді өз күшімен де орындауына болады. Бұл жағдайда негізгі программаны” басында үзу векторды” жүйелі мәнін сақтау керек (мысалы, old\_09 екі сөзді ұяшықта). Ал программаны” со”ында векторды өз қалпына келтіреміз.

```

main      proc
; жүйелік векторды программаны” басында сақтау
mov       AH,  35h        ; векторды алу функциясы
mov       AL,  09h        ; вектор нөмірі
int       21h

```

```
mov    word ptr old_09, BX    ; жүйелі ө"деуішті"
                                салыстырмалы адресі
mov    word ptr old_09+2,ES   ; жүйелі ө"деуішті" сегменті
...
; жүйелік векторды программаны" со"ында қалпына келтіру
push    DS                    ; DS сақтау
lds     DX, old_09            ; old_09-дан DS:DX жазылуы
mov     AH, 25h               ; векторды орнату функциясы
mov     AL, 09h               ; вектор нөмірі
int     21h
pop     DS                    ; адрестеуді қалпына келтіру
...
main    endp
; деректер
old_09 DD    0
```

Ақырында көбінесе жүйелік алгоритмді шамалы өзгерту немесе толықтыру керек. Бұл жағдайда басқа әдістер қолданылады. Жүйелі ө"деуішпен тіркелетін қолданбалы ө"деуішті инициализациялау кезінде, жүйелік ө"деуішті" адресіне көшу мүмкіндігі болу үшін оны" адресін сақтау керек (мысалы old\_09 екі сөзді ұяшықта) және үзу векторына қолданбалы ө"деуішті" адресін (мысалы new\_int) орнату керек. Егер қолданбалы ө"деуіш жүйелік ө"деуішті" со"ынан орындалу керек болса, онда қолданбалы ө"деуіш құрылымы келесі түрде болады:

```
new_int    proc
            pushf
            call CS:old_int ; жүйелік ө"деуішке қайта оралу мен
            ...             ; қолданбалы ө"деу
            iret
new_int    endp
```

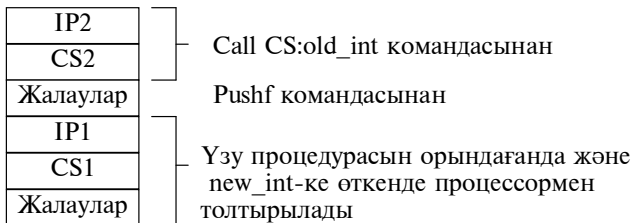
Процессор үзу процедурасын орындап болған со", үзілген процесс стегінде үш сөз болады: жалаушалар сөзі, үзілген программаға екі сөзді қайтару адресі.

IRET командасы 3/4зілген процеске басқаруды қайта беру үшін, деректерді" осындай құрылымы стек төбесінде болу керек. PUSH – жалаулар с/4зін стекке жіберетін 9олданбалы 1/4"деуішті" бірінші командасы. CALL CS:old\_int (м/2нда1ы old\_int DD

операторы қолданбалы екі сөз арқылы берілген) – алыстан шақыру процедурасын командасы. Жүйелік қолданбалы "деуішке басқаруды бере отырып, қолданбалы "деуішті" келесі командасын айту адресін (2 сөз) стекке орналастырады. Соны нәтижесінде IRET командасына қажет, стекте үш сөздік құрылым қалыптасады (5.5-сурет).

Аталған үзуді өтей отырып, жүйелік өдеуіш IRET командасымен аяқталады. Бұл команда стектен үстіңгі үш сөзді алып CS2:IP2 адресі бойынша қолданбалы іске асырады, яғни қолданбалы "деуішті" жолымысы арқылы қайта жолға қояды.

IRET өдеуішінің ақырғы командасы стектен үш жоғарғы сөзді оқып, басқаруды CS1:IP1 адресіне береді.



5.5-сурет. Қолданбалы үзу өдеуіштің орындалу процесінде үзілген программаның стеі

CS1 - үзілген процесстің сегменттік адресі;

IP1 - үзілген процеске қайту нүктесінің салыстырмалы адресі;

CS2 - қолданбалы өдеуіштің сегмент адресі;

IP2 - қолданбалы өдеуішке қайту нүктесінің салыстырмалы адресі.

Егер қолданбалы өдеуіш, жүйелі өдеуішті алдында орналасуы керек болса, онда оны құрылымы келесі түрде болады:

```
new_int:
; қолданбалы өдеу
...
jmp     CS:old_int .
```

Соңғы көшу командасы басқаруды жүйелік өдеуішке береді.

Кейбір кезде қолданбалы ө”деуіш қайсы бір әрекеттерді басқаруды жүйелі ө”деуішке беру алдында немесе берген со” орындайды. Ө”деуіш құрылымы келесі түрде болады:

```
new_int      proc
...           ; жүйелік ө”деуге дейін қолданбалы ө”деу
pushf
call CS:old_int ; жүйелік ө”деуішке қайтару мен
...           ; жүйелік ө”деуден кейінгі қолданбалы ө”деу
iret
new_int      endp
```

Үзулер нәтижесінде басқаруды қолына алған қолданбалы ө”деуіш, ағымды жағдайын (ситуация) талдайды және талдауды” нәтижесіне қарай басқаруды жүйелік ө”деуішке береді немесе жүйелік ө”деуішті келген үзуді қамтамасыз етуден түгел алып тастап, ө”деуді өзі орындайды.

Онда қолданбалы ө”деуішті” құрылымы, жоғарыда келтірілген тәсілдерді” комбинациясынан тұрады:

```
new_int proc
...           ; күйді талдау, егер жүйелік ө”деу
               ; керек болса:
      jmp sys           ; егер жүйелік ө”деу керек болмаса:
...           ; қолданбалы ө”деу
      iret
sys:      jmp CS:old_int
new_int  endp
```

### **5.3. <Ctrl>/C және <Ctrl>/<Break> бойынша үзулерді ө”деу**

<Ctrl>/C пернелер тіркесі белсенді программаны мәжбүрлі түрде аяқтау және басқаруды жүйеге беру үшін арнайы сақталған.

MS-DOS кіріс ағында <Ctrl>/C (ASCII кодасы — 03h) бар болуын пернетақтадан үзулерді ө”деу программасында емес, одан жоғары де”гейде, программалық сұратуларды орындағанда тексереді.

DOS-ты” барлық функциялары екі топқа бөлінеді: 01h...0Ch нөмірлері бар енгізу-шығару функциялары және 00h, 0Dh...6Ch нөмірлері бар қалған басқа функциялар. 6Ch-н үлкен нөмірлі функцияларды” DOS ке”ейтулері, тораптық программалар,

саймандық (инструментальный) пакеттер және басқа да "жүйе жанындағы" программалар пайдаланады. Бұл аталған екі топты" айырмашылығы мынада: енгізу-шығару функцияларын шақырғанда DOS енгізу-шығаруды" ішкі стегіне, ал басқа қалған функцияларды шақырғанда, дискілік деп аталатын басқа ішкі стекке ауысады. DOS-та екі ішкі стекті" болуы оны" жартылай реентерлігін (reenter-қайта кіру мүмкіндігі) қамтамасыз етеді. Яғни ол дегеніміз "дискілік" топтағы DOS функцияларыны" кез келгеніні" орындалу кезінде қате болса, онда енгізу-шығару функциясын шақырып алып, экранға апаттық (аварийный) хабарды шығарып және пернетақтадан пайдаланушыны" жауаптарын енгізу мүмкіндігін қамтамасыз етуі деген сөз.

Енгізу-шығару функцияларыны" (01h...0ch) көпшілігі орындалу алдында пернетақтаны" сақиналы буферінде 03 (<ctrl>/c) кодасы бар ма жоқ па соны тексереді де, егер ол бар болса, онда int 23h командасын орындайды. Әдетте, бұл векторда, ағымдағы процесті аяқтайтын DOS программасыны" адресі болады. Бұған кірмейтін функциялар қатарына 06h және 07h сияқты функциялар, сонымен қатар онда <ctrl>/c бар ма жоқ па деген әрбір 64 шақыруды" біріне сақиналы буферде талдау жасайтын 02h және 09h функциялары болады. (<ctrl>/c) – бар ма жоқ па? Соны тексеру енгізу-шығаруды қайта бағыттаудан бөлек және BREAK жүйелік жалау күйінен бөлек іске асып отырады.

"Дискілік" функциялар (<ctrl>/c) әсер етуді мына жағдайда тексереді: егер BREAK жүйелік жалауы тағайындалған болса (BREAK ON DOS командасы орындалғанда).

Осылайша, BREAK күйін өзгерту арқылы көптеген DOS командаларыны" <ctrl>/c-ге әсер ету механизмін іске қосып, не қоспай отырып басқаруға болады. <ctrl>/c-ны басу арқылы, есептеліп жатқан есепті аяқтауға болмайды. Бұл пернелерді басу арқылы жүйелік функцияларды" шақыруы бар есептерді аяқтауға болады.

Кемшілігі:

<Ctrl>/C пернесін басқанда аяқталған есептердегі өзгертуге ұшыраған ұзу векторлары қалпына келмеуі мүмкін немесе ашық тұрған файлдар дұрыс жабылмауы мүмкін.

Сондықтан көптеген қолданба программалар <Ctrl>C жүйелік ө"деуішті пайдаланбайды. Ол үшін олар өз ө"дегіштерін қолданады.

DOS есепті жадыға жүктегенде PSP белгілі бір ұяшықтарына 22h (есепті аяқтау), 23h (<Ctrl>/C) және 24h (критикалық қатені

ө”деу) векторларыны” мазмұнын көшіріп жазады. Есепті аяқтауды” стандартты жүйелік процедурасы өз құрамында жоғарыда аталған, PSP алынатын үш векторды” мазмұнын ұстайды (сақтайды).

Программалық ө”деуіште, IRET командасын орындамай-ақ, программаны” кез-келген жеріне өтуді қарастыруға болады. Ол (<Ctrl>/C) кез-келген уақытта іске қосуды ұйымдастыруға мүмкіндік береді. Яғни программа орындалу барысын бақылап, оны” корректі түрде аяқталуын қамтамасыз етеді. Бірақ <Ctrl>/C пернесін басқаннан бірден орындалу болмайды. Ол DOS-ты” кез-келген функциясы орындалған со” ғана іске асады.

IBM PC түріндегі машиналарда программа орындалу барысында, кірісуді” екінші мүмкіндігі қарастырылған. Ол (<Ctrl>/<Break>) пернелерін басу арқылы программаға араласу.

BIOS құрамына кіретін пернетақта арқылы үзулерді” жүйелік ө”деуші (<Ctrl>/<Break>) пернелеріні” басуы іске асқанда, адресі 1Bh векторында орналасқан программаға басқаруды береді. Бұл программа бір IRET командасынан тұрады. Бірақ DOS операциялық жүйесіні” бастапқы жүктеу процесінде 1Bh векторыны” мазмұны өзгереді, яғни онда өз ө”деушіні” адресі жазылады. Бұл ө”деуіш басқаруды өз қолына алған со”, мынадай іс-әрекеттер жасайды:

- бас та”баны” орнына пернетақтаны” сақиналы буферінде 0000h іске қосады;

- буфер жүйеге таза қалпында көріну үшін сақиналы буферді” нұсқағышын өзгертеді;

- BIOS деректер аумағыны” ұяшығына 40h:71h адресі бойынша <Ctrl>/<Break> жалауын жазады;

- CON консоль драйверіні” буферіне <Ctrl>/C (03h) кодасын жазады, яғни бұл жағдайда драйвер <Ctrl>/C пернесі басылған деген “ойда” болады.

<Ctrl>/<Break> пернелерін басу нәтижесінде DOS басқаруды 23h векторына береді. Осылайша <Ctrl>/<Break> және <Ctrl>/C жүйелік ө”деулері бірдей қызмет атқарады. <Ctrl>/C ө”деуіні” ерекшелігі:

егер <Ctrl>/C іске қоспай тұрып кейбір пернелер басылса және оларды” кодалары пернетақтаны” сақиналық буферінде қалса, онда олар <Ctrl>/C кодасын “жасырады” яғни консоль драйвері е” ескі та”баны талдайды. <Ctrl>/<Break> - DOS программасына енгізгенде (ол 1Bh үзуді ө”дейді) ол 03h кодасын пернетақтаны” сақиналық буферіне емес, CON драйверіне

жібереді. Сондықтан <Ctrl>/<Break> басуын пернетақтадан та”балар енгізу арқылы жасыруға болмайды (сонымен бірге <Ctrl>/<Break>), ө”дейтін программа сақиналық буферді ө”дейді. Қолданбалы программа 1Bh векторыны” мазмұнын, өз ө”деуішіні” адресімен өзгерте алады. Дегенмен бұл ө”деуішті” программаны аяқтауы IRET командасынан бөлек болады. Ол жүйені авариялық түрде аяқтауға әкелуі мүмкін. Сонымен бірге бұл ө”деуіштен DOS функцияларына тікелей қатынау мүмкін емес.

1Bh векторыны” бастапқы мазмұнын (23h векторынан бөлек), программа аяқталғанда, жүйе автоматты түрде қалпына келтіре алмайды. Сондықтан бұл векторды” бастапқы мазмұнын қалпына келтіру мәселесі туындайды.

Бірақ пайдаланушы программасын авариялық түрде аяқтай алады. Ол үшін ол <Ctrl>/C басып қалпымен аяқтау жолын атап өтеді.

Сондықтан 1Bh векторын ұстап алатын программада өз үзін қамтамасыз ететін ө”деуіш болғаны жөн. Яғни, онда программа аяқталғанда 1Bh векторы қалпына келуі керек.

## **6. РЕЗИДЕНТТІ ПРОГРАММАЛАР**

### **6.1. Резидентті программаларды” ұйымдастырылуыны” негіздері**

Есептеу жүйені” жұмыс істеуін қамтамасыз ететін программаларды” үлкен класы (құрылғыларды” драйверлері, DOS-ты” қоршаулары, деректерді қорғау және шифрлау программалары, русификаторлар, электрондық блокноттар немесе калькуляторлар және т.с.с. қызмет ететін программалар) әр уақытта жадыда болуы және пайдаланушыны” сұратуына немесе есептеу жүйесінде пайда болатын қандай да оқиғаларға бір сәтте әсер етуі керек.

Осындай программалар резидентті деп аталады (Terminate and Stay Resident, TSR). Резидентті программаны” көлемі ықшам (кіші) болу керек, көбінесе программаны” типі .COM болады.

Жадыға жүктеуге және жадыға қалдыруға арналған программалар екі бөлімнен тұрады:

- 1) инициализациялайтын;
- 2) резидентті.

Программаны” мәтінінде резидентті бөлім программаны” басында, ал инициализациялайтын бөлім содан кейін орналасады. Бірінші шақыруда программа жадыға толығымен жүктеледі де, басқару инициализациялайтын бөлімге беріледі. Ол ұзу векторларын толықтырады немесе модификациялайды, программаны нақты бір жұмыс шарттарына бағыттайды (командалық жолда көрсетілген параметрлерді де қолдануы мүмкін) және DOS-ты” int 21h үзуіні” 31h функциясы көмегімен программаны” резидентті бөлімін жадыда қалдыра отырып аяқтайды. Резидентті программаны” көлемі (параграфпен берілген) DX регистрінде беріледі.

Резидентті бөлімні” көлемін анықтау үшін оны келесі операторымен аяқтауға болады:

ressize-\$-muproc

мұнда, muproc - программа басыны” ығысуы, 31h функциясыны” шақырылуы кезінде DX регистріне өрнекті” есептелу нәтижесі жіберіледі:

(ressize+10Fh)/16

31h функциясы резидентті программаға тиісті жады бөліп береді де, басқаруды command.com командалық процессорына береді және есептеу жүйесі бастапқы күйге ауысады. Резидентті программа белсенді болуы үшін оған басқаруды және параметрлерді (егер болса) беру керек.

Резидентті программаны іске қосуды” үш тәсілі бар:

- 1) оны ішкі программа ретінде CALL операторымен шақыру;
- 2) асинхронды (аппараттық) үзулер механизмін қолдану;
- 3) синхронды (программалық) үзулер механизмін қолдану.

Мынадан басқа, DOS-та арнайы резидентті программамен өзара әрекеттесу үшін 2Fh мультиплексті үзуі ұсынылған.

Бірінші тәсіл бойынша ағымдағы белсенді программа жадыда болуы керек.

Екінші тәсіл - сыртқы үзумен резидентті программаны асинхронды жандандыру. Таймерден, пернетақтадан, тізбекті порттан немесе басқа шалғай жабдықтан - сондай сыртқы құрылғыдан үзулері жүйелі және қолданбалы резидентті программалар ке”інен қолданады: принтер спулерлер, календарь, сағат программалары, русификаторлар, калькуляторлар, электронды бланкноттар, электронды анықтамалықтар және деректер қорлары.

Үшінші тәсіл - INT командасыны” көмегімен резидентті программаға басқаруды” синхронды берілуі DOS және BIOS функцияларыны” арасындағы өзара қатынастарды” негізі болып табылады.

Резидентті программаны” құрылымын және оны жадыда қалдыратын жүйелік құралдарын қарастыру.

```

Text          segment      'code'
assume        CS:text, DS:text
org           100h
myproc        proc          far
              jmp          init ; инициализациялайтын секцияға өту
; программаны” резидентті секциясыны” мәліметтері
              . . .
entry:
              . . .          ; программаны” резидентті секциясыны”
                              ; мәтіні
              ired
myproc        endp
ressize equ   $-myproc ; резидентті секцияны” байт бойынша
                              ; көлемі

init          proc          ; инициализация секциясы
              . . .
mov           DX, (ressize+10Fh)/16 ; параграф бойынша көлем
mov AX, 3100h ; аяқтау және жадыда қалдыру функциясы
int 21h
init          endp
text          ends
end           myproc

```

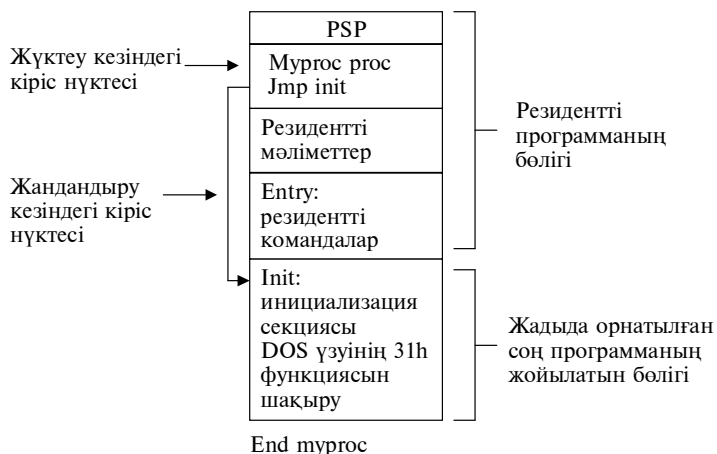
Программаны пернетақтадан бірінші рет іске қосқанда басқару мурпрос процедурасыны” басына беріледі (программалық сегментті” префиксінен кейінгі 1-ші байт). JMP командасыны” көмегімен инициализация секциясына ауысу орындалады. Ол программаны келешекте резидентті болуға дайындайды. Секцияны” со”ғы жолдарында 31h функциясы шақырылады.

Жадыны үнемдеу мақсатында инициализация секциясы программаны” со”ында орналасады және программа аяқталғанда бұл секция жойылады.

ENTRY белгісімен басталатын резидентті программаны” мазмұнды бөлігі аппаратты немесе программалық үзулерді” көмегімен немесе CALL командасыны” көмегімен жандандырылады.

Со”ғы жағдайда резидентті секцияны” программасы RET командасымен (IRET орнына) аяқталады.

6.1-суретте резидентті программаны” құрылымы берілген.



6.1-сурет. Резидентті программаның құрылымы

Резидентті программада екі кіріс нүктесі бар. Программаны жадыға жүктеген со”, басқару программа мәтінін аяқтайтын end операторда көрсетілген нүктесі беріледі (суретте - myproc процедураны” басталуы). Программаны” жүктелу кезінде инициализациялайтын секцияға өту командасы, программаны” бірінші командасы болып табылады (jmp init). Резидентті программа жадыда орнатылған со” пассивті түрінде қала береді. Программаны жандандыру кезінде екінші кіріс нүктесі (entry) қолданылады.

## 6.2. Резидентті программамен байланыс

Транзитті программадан резидентті программаға қатынас жасау үшін есепаралық байланыстар аймағын қолдану керек. Ол аймақ BIOS деректеріні" бір бөлігі болып келеді және 40h:F0h...40h:FFh адрестері бойынша орналасады. Операциялық жүйе бұл аймақты қолданбайды, ол қолданушылар программаларыны" арасында байланыс орнату үшін берілген. Есепаралық байланыстар аймағыны" көмегімен резидентті программаны шақыруға және параметрлерді беруге болады. Параметрлерді" өздерін емес оларды" адрестерін берген ы"ғайлы. Өйткені толық адрес екі сөздік орын алады. Есепаралық байланыстар аймағына төрт адрес сияды.

Инициализациялау секциясыны" орындалу барысында резидентті программа есепаралық байланыстар аймағына өзіні" кіріс нүктесін орналастырады. Мысалы, 40h:F0h сөзіне та"басы entry команданы" салыстырмалы адресі, ал 40h:F2h - CS сегменттік регистріні" мәні.

Транзитті программа басқаруды резидентті программаға беру үшін ES регистрді BIOS деректер аймағыны" басына баптайды және алыс шақыру командасын орындайды.

CALL dword ptr ES:0F0h.

Бұл жағдайда резидентті программа алыс процедура (turproc proc far) ретінде жариялануы керек және ret командасымен аяқталуы тиіс.

Егер резидентті программаға параметрді беру керек болса, оларды" алғашқы адресі транзиттік программамен есепаралық байланыстар аймағыны" басқа ұяшығына жазылады (мысалы, 40h:F4h екі сөзді ұяшығына). Бұл жағдайда резидентті программа, мысалы, SI регистріне 40h:F4h ұяшығынан алынған параметрлерді" салыстырмалы адресін жібереді, ал DS регистріне 40h:F6h ұяшығынан алынған параметрді" сегменттік адресін және бұдан со" параметрлерді" өзін келесі командалармен алады:

```
mov    AX, DS:[SI]
mov    BX, DS:[SI+2] . . .
```

Резидентті программа орындалу алдында барлық қолданылатын регистрлерді сақтау керек, ал программаны" аяқталар алдында регистрлерді" барын қайтадан қалпына келтіру керек.

Резидентті программаға басқаруды және параметрлерді беру кезінде бос векторларды (60h-66h) қолдану.

Инициализация процесінде резидентті программа өзіні” адрестерін бос векторға орналастырады. Мысалы,

```
mov     AX, 0
mov     ES, AX
mov     ES:180h, offset entry ; 60h-адрес
mov     ES:182h, CS.
```

Резидентті программаны жандандыру үшін транзитті программада `int 60h` командасын орындау керек. Мұндай жағдайда резидентті программа `iret` командасымен аяқталуы тиіс. Параметрлер адрестерін басқа да бос векторлар (61h, 62h) арқылы беруге болады.

Егер резидентті программа пернетақтадан іске қосылатын болса, онда оны” `entry` кіріс нүктесіні” адресі пернетақтаны” 09h үзу векторына орналасуы керек. Мұндай кезде пернетақтаны” жүйелі үзу ө”деуішімен қайшылықтар пайда болуы мүмкін.

Іске асатын резидентті программа көптеген аппараттық және программалық үзулерді ұстай алады және оларда бірнеше жандандыру кіріс нүктелері бар.

Параллельді өтіп жатқан процестермен, соны” ішінде резидентті программамен өзара байланыс жасау үшін MS DOS жүйесінде 2Fh үзуіні” стандартты интерфейсі қарастырылған. Бұл үзуді шақыруды” алдында AH регистріне - функция нөмірін, ал AL регистріне - ішкі функция нөмірін жазу керек. 00h...BFh функциялар DOS-қа, ал C0...FFh қолданбалы программаларға арналған. Егер 2Fh үзу көмегімен резидентті программаға параметрлерді” көп санын жіберу немесе алу қажет болса, онда DS:DX немесе DS:SI регистрлерді пайдалануға болады және параметрлер тізімдеріні” адрестерін осылар арқылы беруге болады.

2Fh үзуі AL регистрде резидентті программаны” күйін қайтарады:

AL=0 - программа орнатылмаған (жадыда резидентті программа жоқ) және оны орнатуға болады;

AL=1 - программа орнатылмаған және оны орнатуға болмайды;

AL=FFh - программа орнатылған және оны екінші рет орнатуды” қажеті жоқ.

Қате пайда болғанда CF жалауын орнату керек, ал AX регистріне қате кодын қайтару керек.

Жадыға жүктелінген программаны қайта орнатуды болдырмау үшін 2Fh үзуі қолданылады, кейде осы үзулер арқылы жүктелінген резидентті программаға тиісті ақпарат беріледі (мысалы, жадыдан алып тастау бұйрығы).

Резидентті қолданбалы программа 2Fh үзуіне жауап беру үшін оған осы үзуді” бір немесе бірнеше функцияларыны” қолданбалы ө”дегіштерін қосу керек. Сонда кез келген транзитті немесе басқа резидентті программада бұл үзуте сәйкес, функциясын шақыруы жүктелген резидентті программамен арақатынасын ұйымдастыруға мүмкіндік береді.

2Fh үзуі бар резидентті программаны” құрылымын қарастырайық.

Осы үзуді” C0h функциясын программамызға бекітеміз.

; инициализация секциясы

; осы үзуді” жүйелік функцияларын қолдану үшін 2Fh жүйелік

; ө”деуішті” адресін сақтау керек

init: mov ah, 35h ; векторды алу функциясы

mov al, 2Fh ; вектор  $n\frac{1}{4}$ мірі

int 21h

mov word ptr old\_2Fh, bx ; old\_2Fh ұяшықта

mov word ptr old\_2Fh+2, es ; 2Fh векторын сақтау

; 2Fh векторына ө”деуішімізді” адресін орнатамыз

mov ah, 25h ; векторды орнату функциясы

mov al, 2Fh ; вектор  $n\frac{1}{4}$ мірі

mov dx, offset new\_2Fh

int 21h

; инициализациялау бойынша басқа әрекеттер

...

; программаны аяқтап DOS-ты” 21h үзуіні” 31h функциясыны”

; көмегімен оны резидентті жадыда қалдыру

...

; деректерді” резидентті өрісі

old\_2Fh DD 0 ; 2Fh векторыны” бастапқы мазмұнына

; сақтауға арналған ұяшық

; жұмыс секция

entry: . . .

iret

; бізді” 2Fh үзу ө”деуіші (C0h функция)

new\_2Fh: cmp ah, 0C0h ; бізді” функциямыз ба?

```

        je      c0h          ; біздікі
        jmp     csold_2Fh    ; бізді” емес, жүйелік ө”деуішке
c0h:
; Al-ді талдау мүмкін және ішкі функцияны іске асыру
; программасына көшу: параметрлерді беру немесе алу, резидентті
; программаны баптау немесе оны жадыдан алып тастау
        . . .
        iret                ; шақырған программаға қайта оралу

```

### 6.3. Қайта орнатуын тексеру

Резидентті программаны қайта жүктеуден сақтау үшін, программаны” жүктемелік модуліне еркін кода (сигнатура) енгізу және программаны” инициализация процесінде сол коданы” жадыда бар болуын бақылап тұру керек. Мұндай әдіс аппаратты ұзу арқылы қосылатын резидентті программалар үшін ы”ғайлы (мысалы, 09h).

Егер бізді” программамыз әлі орнатылмаған болса, онда 09h векторында сигнатурасы жоқ ұзу ө”дегішіні” адресі сақталған. Ал егер программа орнатылған болса, жадыда оны” сигнатурасы болады; қайта инициализация процесі оны тауып программаны апатты аяқтайды.

Өзін қайта жүктелуден сақтайтын программаны” құрылымы келесі түрде болуы мүмкін:

```

        org     100h          ; бізді” com-программамыз
myproc proc
        jmp     init          ; инициализациялауға жіберу
mark    dw      1234h         ; сигнатура
entry:
; жұмысшы секция
        . . .
        iret
init:
; инициализация секциясы
        mov     ax, 3509h      ;бұл ө”дегіш пернетақтадан болсын
        Int21h                ; ES:BX= 09h векторды” мазмұны
        sub     bx, 2          ; егер бізді” программа жүктелген
                                ; болса BX --> entry-2, яғни mark
        cmp     word ptr ES:[BX], 1234h
        je      installed     ; сигнатура табылды

```

; сигнатура табылмады, инициализацияны ,pi қарай жалғастыру  
; программаны жадыға орнату

...

installed:

; “программа орнатылды” деген хабарды шығару

...

```
mov    ax, 4C01h    ; қате кодымен және жадыда
                    ; қалдырмай аяқтау
```

```
int     21h
```

Енді бұл программа орнатылған со”, жадыға басқа резидентті программаны” 09h үзу ө”деуіші жүктелсе онда жоғарыда берілген әдіс қайта жүктелуден сақтамайды. Өйткені үзу векторында сигнатурасы жоқ басқа программаны” адресі болады.

Резидентті программаларды қайта жүктеуден сақтау үшін 2Fh мультиплексті үзуін пайдалану қажет. Осындай тәсілмен қорғайтын резидентті программаны” құрылымы келесі түрде болуы мүмкін.

; Инициализация секциясы

init:

; бұл программаны” жадыға жүктелуін тексеру

```
mov    ax, 0C000h    ; C0h- функция, 00h- ішкі функция
int     2Fh           ; ө”деушіміз al= FFh қайтару керек
cmp     al, 0FFh      ; орнатылған?
je      installed     ; ия- апатты аяқталуына
```

; Программа орнатылмаған. Оны” инициализациялануына кірісеміз

; 2Fh жүйелік ө”деушіні” адресін сақтаймыз

```
mov     ah, 35h       ; векторды оқу функциясы
```

```
mov     al, 2Fh       ; вектор нөмірі
```

```
int     21h
```

```
mov     word ptr old_2Fh, BX ; old_2Fh ұяшықта 2Fh
```

векторды

```
mov     word ptr old_2Fh+2, ES ; сақтаймыз
```

; 2Fh векторына бізді” ө”деуіш адресін жүктейміз

```
mov     ah, 25h       ; векторды орнату функциясы
```

```
mov     al, 2Fh       ; вектор нөмірі
```

```
mov     dx, offset new_2Fh
```

```
int     21h
```

```
; инициализацияны” жалғасуы: басқа векторларды жүктеу,
программаны
; орнатуы және онымен жұмыс істеу ережелері туралы хабар
шығару
. . .
; программаны жадыда қалдыру
    mov     ax, 3100h
    mov     dx, (resize+10Fh)/16
    int     21h
new_2Fh   proc
; программаны” резидентті секциясыны” құрамына кіретін 2Fh
ө”деуіші
    cmp     ax, 0C000h      ; бізді” функция және ішкі функция?
    jne     notour          ; жоқ, 2Fh жүйелік үзу ө”деуішін
                                ; шақырамыз
    mov     al, 0FFh        ; программа жадыға жүктелген - FFh
                                ; кодасын қайтарамыз
    iret                                ; ө”деуіштен шығу
notour: jmp cs:old_2Fh      ; 2Fh жүйелік үзу ө”деуішіне өту
; Деректерді” резидентті өрістері
old_2Fh    dd      0        ; 2Fh векторды” бастапқы мазмұнын
                                ; сақтауға арналған ұяшық
```

## А ҚОСЫМШАСЫ

### ПРОЦЕССОРДЫҢ КОМАНДАЛАРЫ

**MOV** — қайта жіберу. Бұл команда регистрды”, жадыны” мәнін немесе оны” тікелей мәнін басқа регистрге немесе жадыға көшіреді. Бірдей өлшемді екі операндысы бар (байт, сөз немесе екілік сөз). Жадыдан жадыға көшіру мүмкін емес (ол үшін мәнді алдымен жадыдан регистрге, содан со” регистрден жадыға көшіру керек).

Мысалы:

```
mov AX, BX           ; BX-тан AX-қа көшіру
mov DS, AX           ; AX-тан DS-қа көшіру
mov EAX, DS:[1234h]  ; жадыдан DS:1234h адресі бойынша
                    ; екілік сөзді EAX регистрге көшіру
mov ESI, ES:[EBX+EAX]; жадыдан ES:EBX+EAX адресі
                    ; бойынша екілік сөзді ESI регистрге
                    ; көшіру
```

**MOVZX** (8086 процессорында жоқ) — санды ке”ейтумен қайта жіберу. Команда санды ке”ейте отырып көшіреді (оған нөлдер қосу арқылы оны” өлшемін үлкейтеді). Екі операндысы бар, біріншісіні” өлшемі екіншісінен үлкен.

Мысал:

```
movzx EAX, BL.
```

Осы операция орындалған со” EAX регистріні” 8-31 разрядтары нөлге те” болады, ал 0-7 разрядтары BL-ға те” болады.

**MOVSX** (8086 процессорында жоқ) — санды та”басы бойынша ке”ейтумен қайта жіберу. Команда санды үлкен разряд бойынша ке”ейте отырып көшіреді. Екі операндысы бар, біріншісіні” өлшемі екіншісінен үлкен.

Мысал:

```
movsx EBX, AX.
```

Осы операция орындалған со” EBX регистріні” 0-15 разрядтары AX-ке те” болады, ал 16-31 разрядтары AX-ты” 15-ші разрядына те” болады.

**XCHG** — алмасу. Командасы регистрлер мәнімен немесе регистр мен жады мәндеріні” орындарын алмастырады. Командада өлшемдері бірдей екі операнды бар.

Мысал:

xchg AH, AL.

**ADD** - қосу. Бұл команда бір санды басқа санмен қосып, нәтижені бірінші санны” орнына жазады. Командада өлшемдері бірдей екі операндысы бар. Команда операция нәтижесінде CF, PF, AF, ZF, SF және OF жалауларын орнатады.

Мысалы:

add AX, BX ; AX=AX+BX.

**ADC** — тасымалды есепке ала отырып қосу. Команда бір санды екінші санға қосады, оған тасымал жалауын қосып, нәтижені бірінші санны” орнына жазады. Командада өлшемдері бірдей екі операндысы бар. Команда операция нәтижесінде CF, PF, AF, ZF, SF және OF жалауларын орнатады.

Мысалы:

adc BX, DS: [1234h] ; BX=BX + (DS:1234h адресі  
; бойынша) сөз + CF.

Бұл команда үлкен сандарды қосқанда тасымалды есепке алу үшін қажет. Мысалы әрбіреуіні” өлшемі сегіз байт орын алатын екі санды қосу керек делік. Бірінші сан EAX (32-63 разрядтары) және EBX (0-31 разрядтары) регистрлерінде болсын, екінші сан ECX (32-63 разрядтары) пен EDX (0-31 разрядтары) регистрлерінде болсын. Келесі командалар тізбегі екі санды қосады.

add EBX, EDX  
adc EAX, ECX.

**SUB** — азайту. Бұл команда бір саннан екінші санды алып, нәтижені бірінші сан орнына қояды. Командада өлшемдері бірдей екі операндысы бар. Команда операция нәтижесінде CF, PF, AF, ZF, SF және OF жалауларын орнатады.

Мысалы:

sub DS:[EBX], AX  
; DS:EBX адресі бойынша сөз = DS:EBX адресі бойынша сөз – AX.

**SBB** – тасымалды есепке ала отырып азайту. Команда бір саннан екінші санды алады, одан тасымал жалауын алып, нәтижені бірінші санның орнына жазады. Командада өлшемдері бірдей екі операнды бар. Команда операция нәтижесінде CF, PF, AF, ZF, SF және OF жалауларын орнатады.

Мысалы:

sbb AX, BX ; AX=AX – BX – CF.

**INC** – инкремент, регистрдің немесе жадының мәніне бір бірлікті қосу. Командада бір операнд бар. Команда операция нәтижесінде PF, AF, ZF, SF және OF жалауларын орнатады.

Мысалы:

inc AX ; AX=AX+1  
inc BH ; BH=BH+1.

**DEC** – декремент, регистрдің немесе жадының мәнін бір бірлікке азайту. Командада бір операнд бар. Операциялардың қорытындыларынан кейін бұл команда PF, AF, ZF, SF және OF жалауларын орнатады.

Мысалы:

dec EBX ; EBX=EBX–1.

**MUL, IMUL** – көбейту (біріншісі таңбаны есепке алмайды, екіншісі таңбаны есепке алады). Команда регистрдің немесе жадыдағы санды AL, AX немесе EAX (сан разрядына байланысты) регистріне көбейтеді. Командада бір операнд бар – регистр немесе жадыдағы сан (екіншісі әрқашанда AL, AX, немесе EAX). Нәтиже – өлшемі операндтан екі есе үлкен болады, ол AX, EAX және DX, EAX пен EDX-те орналастырылады. Команда операция орындалу нәтижесінде CF және OF жалауларын орнатады. PF, AF, ZF, және SF жалаулары анықталмаған.

Мысалы:

mul BX ; DX пен AX =AX\*BX.

**DIV, IDIV** – бөлу (біріншісі таңбаны есепке алмайды, екіншісі таңбаны есепке алады). Команда AX, EAX және DX немесе EAX және EDX регистрлердегі санды, регистрге немесе

жадыдағы санға бөледі, нәтиже AL регистрінде (қалдық AH-та), AX-та (қалдық DX-та) немесе EAX-та (қалдық EDX-та) болады. PF, AF, ZF, SF, CF және OF жалаулары анықталмаған.

div BX ; (DX пен AX) : BX = AX.

**CMP** – салыстыру. Бұл команда регистрді регистрмен, жады мәнімен немесе тікелей мәнмен салыстырады. Команда бір саннан екіншісін алып тастайды да, нәтижеге байланысты жалауларды шығарады, бірақ нәтижені сақтамайды. Командада өлшемдері бірдей екі операнды бар. Команда операция нәтижесінде CF, PF, AF, ZF, SF және OF жалауларын орнатады.

Мысалы:

cmp AX, BX ; Жалаулар AX–BX нәтижесіне байланысты ; орнатылады.

**AND** – логикалық ЖӘНЕ. Егер екі операндты” разрядында бірлік бар болса, онда команда нәтижесінде алынған сан разрядына бірлік жазылады (бірінші операнд ЖӘНЕ екінші операнд разрядыны” бірлігі бар). Командада өлшемдері бірдей екі операндысы бар. Команда операция нәтижесінде PF, ZF және SF жалауларын орнатады. AF жалауы анықталмаған. CF, OF жалаулары босатылады.

Мысалы:

and AL, BL; AL=AL AND BL, егер мысалы ; AL=10001010b болса, ал BL=10010010b, ; онда нәтиже 10000010b те”.

**OR** – логикалық НЕМЕСЕ. Егер е” болмағанда бір операндты” разрядында бірлік бар болса, онда команда нәтижесінде алынған сан разрядына бірлік жазылады (бірінші операнд НЕМЕСЕ екінші операнд разрядыны” бірлігі бар). Командада өлшемдері бірдей екі операнды бар. Команда операция нәтижесінде PF, ZF және SF жалауларын орнатады. AF жалауы анықталмаған. CF, OF жалаулары босатылады.

Мысалы:

or CL,DH; CL=CL OR DH, егер мысалы ; CL=10001010b болса, ал DL=10010010b, ; онда нәтиже 10011011b те” болады.

**XOR** — шығарушы НЕМЕСЕ (модуль 2 бойынша қосу). Егер екі операндта разрядты" мәндері бірдей болса, онда команда нәтижесінде алынған санны" разрядына ноль жазылады, ал бірдей болмаса - бірлік жазылады. Командада өлшемдері бірдей екі операнды бар. Команда операция нәтижесінде PF, ZF және SF жалауларын орнатады. AF жалауы анықталмаған. CF, OF жалаулары босатылады.

Мысалы:

xor AH, AL ; AH=AH XOR AL, егер мысалы  
; AH=10001010b болса, ал AL= 10010010b,  
; онда нәтиже 00011000b те" болады.

**TEST** — разрядтарды тексеру. Команда логикалық ЖӘНЕ орындайды және операция нәтижесінде жалауларды шығарады, бірақ нәтижені сақтамайды. Командада өлшемдері бірдей екі операнды бар. Команда операция нәтижесінде PF, ZF және SF жалауларын орнатады. AF жалауы анықталмаған. CF, OF жалаулары босатылады.

Мысалы:

test AX, 0001h  
test ECX, 0000007Fh.

**ROL** — солға циклдік ығысу (ротация). Команда разрядтарды солға ығыстырады, үлкен разряд кішіге түседі. Команданы" екі операндысы бар: ығыстыруға қажет регистр немесе жады мәні және санды ығыстыруға қажет разряд саны. Бұл сан ретінде тікелей сан немесе CL регистрі болады. Үлкен разрядтан кішіге тасымалдана алатын разряд CF тасымалдау жалауына түседі. Операция орындалу нәтижесінде CF жалауы орнатылады, OF анықталмаған, қалған жалаулар өзгермейді.

Мысалы:

rolAL, 1.

Егер, мысалы AL=11001000b те" болса, онда операциядан кейін ол 10010001b те" болады.

**ROR** — о"ға циклдік ығысу. Команда разрядтарды о"ға ығыстырады, кіші разряд үлкен разрядқа ауысады. Команданы" екі операндысы бар: ығыстыруға қажет регистр немесе жады мәні және санды ығыстыруға қажет разряд саны. Бұл сан ретінде тікелей сан немесе CL регистрі болады. Кіші разрядтан үлкенге тасымалдана алатын разряд CF тасымалдау жалауына түседі.

Операция орындалу нәтижесінде CF жалауы орнатылады, OF анықталмаған, қалған жалаулар өзгермейді.

Мысалы:

rotAL, 1.

Егер, мысалы AL=11001000b те” болса, онда операциядан кейін ол 01100100b те” болады.

**RCL** — тасымал арқылы солға циклдік ығысу. Команда разрядтарды солға ығыстырады, үлкен разряд тасымалдау жалауына түседі, ал ол кіші разрядқа жылжиды. Операндтар мен жалауға әсерлері ROL командасындай.

**RCR** — тасымал арқылы о”ға циклдік ығысу. Команда разрядтарды о”ға ығыстырады, кіші разряд тасымалдау жалауына түседі, ал ол үлкен разрядқа жылжиды. Операндтар мен жалауға әсерлері ROR командасындай.

**SHL** — солға логикалық ығысу. Команда разрядтарды солға ығыстырады, үлкен разряд тасымалдау жалауына түседі, ал кіші разрядқа ноль түседі. Бір разрядқа жылжу (ығысу) 2-ге көбейткенге те”. Операндтары циклдік ығысу командаларындағыдай. Команда операция нәтижесінде CF, PF, ZF және SF жалауларын орнатады, AF және OF жалаулары анықталмаған.

**SHR** — о”ға логикалық ығысу. Команда разрядтарды о”ға ығыстырады, кіші разряд тасымалдау жалауына түседі, ал үлкен разрядқа ноль түседі. Бір разрядқа жылжу (ығысу) 2-ге бөлуге те”. Операндтары циклдік ығысу командаларындағыдай. Команда операция нәтижесінде CF, PF, ZF және SF жалауларын орнатады, AF және OF жалаулары анықталмаған.

**SAL** — солға арифметикалық ығысу. Команда разрядтарды солға ығыстырады, үлкен разряд тасымалдау жалауына түседі, ал кіші разрядқа ноль түседі. Бір разрядқа ығысу та”балы санды екіге көбейтуге те”. Операндтары циклдік ығысу командаларындағыдай. Команда операция нәтижесінде CF, PF, ZF және SF жалауларын орнатады, AF жалау анықталмаған және OF жалау алып тасталған.

**SAR** – о”ға арифметикалық ығысу. Команда разрядтарды о”ға ығыстырады, кіші разряд тасымалдау жалауына түседі, ал үлкен разряд өзгеріссіз қалады. Бір разрядқа ығысу та”балы санды екіге бөлуге те”. Операндттары циклдік ығысу командаларындағыдай. Команда операция нәтижесінде CF, PF, ZF және SF жалауларын орнатады, AF жалау анықталмаған және OF жалау алып тасталған.

**SHLD** – (8086 процессорында жоқ) логикалық түрдегі солға қос ығысу. Команда разрядтарды солға бір регистрден екіншісіне ығыстырады. Бірінші регистр мәні өзгермейді. Екіншісіні” разрядтары солға ығысады. Команданы” үш операндасы, екі регистрі немесе жадыны” мәні және ығысу шамасы бар. Екі операндты” разрядтылығы те”, ығысу шамасы басқа. Команда операция нәтижесінде CF, PF, ZF және SF жалауларын орнатады, AF және OF жалаулары анықталмаған.

**SHRD** - (8086 процессорында жоқ) логикалық түрдегі о”ға қос ығысу. Команда разрядтарды о”ға бір регистрден екіншісіне ығыстырады. Бірінші регистр мәні өзгермейді. Екіншісіні” разрядтары о”ға ығысады. Команданы” үш операндасы, екі регистрі немесе жадыны” мәні және ығысу шамасы бар. Екі операндты” ығысу шамасынан басқа разрядтылығы те”. Команда операция нәтижесінде CF, PF, ZF және SF жалауларын орнатады, AF және OF жалаулары анықталмаған.

**IN** – порттан оқу. Команда байтты немесе сөзді порттан оқиды. Порт адресі DX регистрінде беріледі немесе команда да болады. Егер сөз оқылса, ол AX регистріне оқылады, (адресі тікелей көрсетілген портынан байт AL регистрге түседі, ал адресі бірге артық порттан AH регистріне түседі), егер байт болса, онда AL регистріне оқылады.

Мысалы:

in AL, DX.

**OUT** – байтты портқа жазу. Команда байтты немесе сөзді портқа жазады, оны” адресі тікелей беріледі немесе DX регистрінде болады. Егер сөз жазылса, ол AX регистрінен жазылады, егер байт жазылса, онда ол AL регистрінен жазылады.

Мысалы:

out DX, AX.

; AL DX портына түседі, AH DX+1 портына түседі.

**LODS** – қатарды оқу. Команда мәнді DS:SI адресі бойынша AL , AX немесе EAX-ке оқиды (енгізеді) және SI регистрін 1, 2 немесе 4-ке арттырады немесе азайтады (DF жалауыны” мәні бойынша). SI регистріні” шамасын арттыру немесе азайту және мәні жазылатын регистрді та”дау команда өлшеміне байланысты болады. Ол өлшемді көрсету үшін мына суффикстер қойылады: B- байт, W- сөз, D-қос сөз.

Мысалы:

lodsw.

**STOS** – қатарға жазу. Команда мәнді AL , AX немесе EAX-тан оқып, ES:DI адресі бойынша қатарға жазады және DI регистрін 1, 2 немесе 4-ке артырады немесе азайтады (DF жалауыны” мәні бойынша). DI регистріні” шамасын көбейту немесе азайту және мәні оқылатын регистрді та”дау команда өлшеміне байланысты болады. Ол өлшемді көрсету үшін мына суффикстер қойылады: B- байт, W- сөз, D-қос сөз.

Мысалы:

Stosd.

**MOVS** – қатарды жіберу. Команда мәнді (сөз және байт немесе қос сөз) DS:SI адресінен ES:DI адресіне жібереді және SI және DI регистрлерін 1, 2 немесе 4-ке көбейтеді немесе азайтады (DF жалауыны” мәні бойынша). SI және DI регистрлеріні” өзгеру шамасы команда өлшеміне байланысты. Ол өлшемді көрсету үшін мына суффикстер қойылады: B- байт, W- сөз, D-қос сөз.

**SCAS** – қатарда мәнді іздеу. Команда мәнді ES:DI адресі бойынша AL, AX немесе EAX регистрлерімен салыстырады және DI регистрін 1, 2 немесе 4-ке көбейтеді немесе азайтады (DF жалауыны” мәні бойынша). Мән жазылатын регистрді та”дау және DI регистріні” көбейту немесе азайту шамасы команда өлшеміне байланысты болады. Ол өлшемді көрсету үшін команда со”ына суффикс қойылады: B-байт, W- сөз, D-қоссөз.

**CMPS** — қатарды салыстыру. Команда мәнді DS:SI және ES:DI адрестері бойынша салыстырады және SI мен DI регистрлерін 1, 2 немесе 4-ке көбейтеді немесе азайтады (DF жалауыны” мәні бойынша). SI және DI регистріні” көбейту немесе азайту шамасы команда өлшеміне байланысты болады. Өлшемді көрсету үшін команда со”ына суффикс қойылады: B-байт, W- сөз, D-қоссөз.

**INS** — порттан қатарларды (жолдарды) енгізу. Команда адрес нөмірі DX регистріндегі порттан мәнді оқып, оны ES:DI адрес бойынша жазып, DI-ді көбейтеді немесе азайтады (DF жалауыны” мәні бойынша).

**OUTS** — портқа қатарларды (жолдарды) шығару. Команда DS:SI адрес бойынша мәнді оқып, оны адрес нөмірі DX регистріндегі портқа жазып, SI -ді көбейтеді немесе азайтады (DF жалауыны” мәні бойынша).

**REP, REPZ (REPE), REPNZ (REPNE)** — қайталау префикстері. Бұл префикстер қатарлар операциялар (LODS, STOS, MOVS, SCAS, CMPS, INS, OUTS) командаларыны” алдында тұруы мүмкін. Егер команда алдында REP префиксі тұрса, онда команда орындалғанда CX регистрі азаяды және де егер ол нөлге те” болмаса, онда команда қайталанады. Егер REPZ префиксі тұрса, онда команда CX регистр нөлге те” болғанға дейін және ZF=1 (нөл жалауы) тұрғанда қайталанып тұрады. Егер REPNZ префиксі тұрса, онда команда CX регистр нөлге те” болмағанға дейін және ZF=0 болғанда қайталанып тұрады.

**JMP** — шартсыз өту. Бұл команда басқаруды программаны” көрсетілген нүктесіне береді. Басқаруды осы сегмент кодасыны” нүктесіне (жақын өту) беруге болады, сонымен қатар басқа сегмент нүктесіне де (алыс өту) беруге болады. Барлық өту командаларында бір операнд-адрес бар, өту сол арқылы іске асады.

**JC (JB, JNAE)** — шартты өту. Егер тасымал жалауы CF=1 болса, өту іске асады (В-төменгі).

**JNC (JNB, JAE)** – шартты өту. Егер тасымал жалауы  $CF=0$  (А-жоғары немесе Е-те”) болса өту іске асады.

**JP (JPE)** – шартты өту. Егер PF жұптық жалауы 1-ге те” болса, онда өту іске асады (жұптық).

**JNP (JPO)** – шартты өту. Егер PF жұптық жалауы 0-ге те” болса, онда өту іске асады (жұптық емес).

**JZ (JE)** – шартты өту. Егер ZF нөлдік жалауы 1-ге те” болса, онда өту іске асады (те”).

**JNZ (JNE)** – шартты өту. Егер ZF нөлдік жалауы 0-ге те” болса, онда өту іске асады (те” емес).

**JS** – шартты өту. Егер SF та”балық жалауы 1-ге те” болса, онда өту іске асады (теріс).

**JNS** – шартты өту. Егер SF та”балық жалауы 0-ге те” болса, онда өту іске асады (о”).

**JO** – шартты өту. Егер OF толып кету жалауы 1-ге те” болса, онда өту іске асады (толып кету).

**JNO** – шартты өту. Егер OF толып кету жалауы 0-ге те” болса, онда өту іске асады (толмаған).

**JBE (JNA)** – шартты өту. Егер CF тасымал жалауы немесе ZF нөлдік жалауыны” тым болмағанда біреуі 1-ге те” болса, онда өту іске асады (төмен немесе те”).

**JNBE (JA)** – шартты өту. Егер CF тасымал жалауы және ZF нөлдік жалауы 0-ге те” болса, онда өту іске асады (жоғары).

**JL (JNGE)** – шартты өту. Егер OF толып кету жалауы мен SF та”ба жалауыны” бірі 1-ге те” болса, ал екіншісі 0-ге те” болса, онда өту іске асады (кіші).

**JNL (JGE)** – шартты өту. Егер OF толып кету жалауы мен SF та"ба жалауы те" болса: екеуі де 1-ге те" болса немесе екеуі де 0-ге те" болса, онда өту іске асады (үлкен немесе те").

**JLE (JNG)** – шартты өту. Егер OF толып кету жалауы не SF та"ба жалауыны" бірі 1-ге те" болса, екіншісі 0-ге те" болса немесе тасымал жалауы 1-ге те" болса, онда өту іске асады (кем немесе те").

**JNLE (JG)** – шартты өту. Егер OF толып кету жалауы мен SF та"ба жалауы өзара те" болса (екеуі де 1-ге те" болса немесе екеуі де 0-ге те") және ZF нөлдік жалауы 0-ге те" болса, онда өту іске асады (үлкен).

**LOOP, LOOPZ (LOOPE), LOOPNZ (LOOPNE)** – цикл. Команда CX регистріні" ішкі мазмұнын азайтады, егер CX нөлге те" болмаса, онда ол көрсетілген адреске өтеді. CX нөлге те" болмағанға дейін және ZF ноль жалауы 1-ге те" тұрғанға дейін LOOPZ командасы орындалады. CX нөлге те" болғанға дейін және ZF нөлдік жалау 0-ге те" тұрғанға дейін LOOPNZ командасы орындалады.

**CALL** – ішкі программаны шақыру. Бұл команда келесі команданы" адресін стекте сақтайды (қайтару адресі) және басқаруды ішкі программаға береді. Осы сегменттегі басқа да ішкі программаларды шақыру мүмкіндігі бар (жақын ішкі программалар). Басқа сегменттегі ішкі программаны (алыс ішкі программа) шақыру мүмкіндігі де бар. Команданы" бір операндасы бар, ол – ішкі программаны" адресі.

**RET** – ішкі программдан қайту. Бұл команда стектен қайту адресін оқиды және басқаруды осы адрес бойынша береді. Команда сонымен бірге *жақын* және *алыс* ішкі программдан басқаруды қайтара алады. Егер ішкі программа CALL *алыс* командасы арқылы шақырылса, онда RET командасы *алыс* (RETF) болуы керек. Егер *жақын* CALL командасы болса, онда RET командасы де *жақын* болады. *Жақын* RET командасы сегменттегі тек ығысуды қалпына келтіреді, ал *алыс* сегментті де, ығысуды да қалпына келтіреді.

**INT** — программаны үзу. Команда **CALL** командасына ұқсайды. Ол стекте қайтару адресін (сегмент және ығысу) және жалаулар регистрін сақтайды және үзуді өтейтін программаны шақырады. Бір операнды бар, ол үзу нөмірі (00h-FFh).

**IRET** — үзуден қайтару (қайту). Команда стектен үзуден қайту адресі мен жалауларды оқиды және басқаруды шақыру программасына береді.

**PUSH** — стекте мәнді сақтау (стекке жазу). Команда стекте регистрді немесе жады мәнін сақтайды. Бұл команда көмегімен тек сөз және қос сөзді сақтауға болады.

Мысалы:

push        EAX.

**POP** — стектегі мәнді қалпына келтіру (стектен оқу). **PUSH** команда арқылы стекте сақталған регистрді немесе жады мәнін қалпына келтіреді. Бұл команда көмегімен тек сөз және қос сөзді қалпына келтіруге болады.

**PUSHF** — **FLAGS** жалаулар регистрді стекте сақтайды. Команда **EFLAGS** регистрді сақтау үшін, команда соңына D-қою керек.

Мысалы:

pushfd.

**POPF** — **FLAGS** жалаулар регистрді стектен қалпына келтіреді. Команда **EFLAGS** регистрді қалпына келтіру үшін, команда соңына D-қою керек.

**PUSHA** (8086 процессорында жоқ) — стекте барлық жалпы мақсатты регистрлерді сақтау. Команда стекте барлық жалпы мақсатты регистрлерді сақтайды (**AX**, **BX**, **CX**, **DX**, **SP**, **BP**, **SI** және **DI**). Команда 32-разрядтық жалпы мақсатты регистрлерді сақтау үшін оны соңында D-суффиксі болу керек.

Мысалы:

pushad

Стекте **EAX**, **EBX**, **ECX**, **EDX**, **ESP**, **EBP**, **ESI** және **EDI** регистрлері сақталады.

**POP** (8086 процессорында жоқ) – стектен барлық жалпы мақсатты регистрлерді қалпына келтіру. Команда стектен барлық жалпы мақсатты регистрлерді қалпына келтіреді (AX, BX, CX, DX, SP, BP, SI және DI). Команда 32-разрядты жалпы мақсатты регистрлерді қалпына келтіру үшін оны” со”ында D-суффиксі болу керек.

**CLC** – CF тасымал жалауын босату (CF=0).

**CLD** – DF бағыт жалауын босату. Бұл команда орындалған со” қатарлық операциялар SI және DI регистрлер мәнін көбейтеді.

**CLI** – IF үзуге рұқсат беру жалауын босату (IF=0). Бұл командадан со” барлық үзулерге тыйым салынады.

**CMC** – CF тасымал жалауын өзгерту (терістеу). Команда тасымал жалауыны” мәнін қарама-қарсы мәнге өзгертеті.

**STC** – тасымал жалауын орнату (CF=1).

**STD** – бағыттау жалауын орнату (DF=1). Бұл команда орындалған со”, қатарлық операциялар SI және DI регистрлер мәнін азайтады.

**STI** – үзуге рұқсат беру жалауын орнату (IF=1). Бұл командадан со” барлық үзулерге рұқсат етіледі.

**SET[шарт]** – мәнді шарт бойынша орнату. Команда шартты” біреуін тексереді (шартты өту командасындағыдай) және егер бұл шарт орындалмаса регистрге ноль орнатады, егер, орындалса онда регистр бірге орнатылады. Командада бір операндысы бар - 8-разрядтық регистр немесе жады мәні.

Мысалы:

setnz AL.

**BT** (8086 процессорында жоқ) – разрядты тексеру. Команда көрсетілген разрядты тасымал жалауына көшіреді. Команданы” екі операндысы бар - регистр немесе жады мәні және регистрде болатын бит нөмірі.

**BTC** (8086 процессорында жоқ) — терістеу арқылы разрядты тексеру. Команда көрсетілген разрядты тасымал жалауына терістеу арқылы көшіреді. Команданың екі операндысы бар - регистр немесе жады мәні және регистрде болатын разряд нөмірі.

**BTS** (8086 процессорында жоқ) — разрядты тексеру және орнату. Команда разрядты тасымал жалауына көшіреді және оны 1-ге орнатады. Команданың екі операндысы бар - регистр немесе жады мәні және регистрде болатын разряд нөмірі.

**BTR** (8086 процессорында жоқ) — разрядты тексеру және алып тастау. Команда разрядты тасымал жалауына көшіреді және алып тастайды. Команданың екі операндысы бар - регистр немесе жады мәні және регистрде болатын разряд нөмірі.

**BSF** (8086 процессорында жоқ) — разрядтарды тура бағытта сканерлеу (іздеу және тексеру). Команда кіші разрядтағы мәндерді кезегімен тексеріп, көрсетілген регистрге бірінші кездескен бірлігі бар разрядты кіргізеді. Егер разрядтарда бірліктер болмаса, онда нөлдік жалауына бірлік орнатылады.

**BSR** (8086 процессорында жоқ) — разрядтарды кері бағытта сканерлеу. Команда үлкен разрядтағы мәндерді кезегімен тексеріп, көрсетілген регистрге бірінші кездескен бірлігі бар разрядты кіргізеді. Егер разрядтарда бірліктер болмаса, онда нөлдік жалауына бірлік орнатылады.

## Ә ҚОСЫМШАСЫ

### DOS ФУНКЦИЯЛАРЫ БОЙЫНША АНЫҚТАМАЛЫҚ МӘЛІМЕТТЕР

#### **INT 21h, 01h функциясы. Символды бейнемен енгізу.**

Стандарттық енгізу құрылғысынан символды енгізу мен оны стандартты шығару құрылғысында бейнелейді. Символ болмаған жағдайда енгізуді күтеді. Енгізуді” қайта бағытталуына рұқсат етіледі. Егер енгізу қайта бағытталмаған болса <Ctrl>/C ө”деуін орындайды. Егер енгізу қайта бағытталған болса BREAK (BREAK=ON) қосылған жағдайында ғана <Ctrl>/C ө”деуін орындайды. Ке”ейтілген ASCII кодын оқу үшін функцияны қайта орындауды талап етеді.

Функцияны шақырғанда: AH=02h.

Функциядан қайтарғанда: AL= енгізу деректерді”  
байты.

#### **INT 21H, 02h функциясы. Символды шығару.**

Символды экран бетіне шығарады. Шығаруды” қайта бағытталуына рұқсат етіледі. <Ctrl>/C ө”деуін әрбір 64-ші символын экранға шығару алдында орындайды. ASCII кодалары: 07h - қо”ырау, 08h - бір қадам артқа жылжу, 09 – табуляция, 0Dh - каретканы қайтару, 0Ah - жолды ауыстыру басқарушы сияқты қарастырылады және өзіне сәйкес әрекетпен орындалады.

Шақырғанда: AH=02h,

DL=деректер байты.

**INT 21h, 05h функциясы. Принтерге символды шығару.**

Символды стандартты принтерге шығарады. Егер принтер бос емес болса, онда функция оны” босатылуын күтеді. <Ctrl>/C ө”деуін орындайды.

Шақырғанда: AH=05h,

DL=деректер байты.

**INT 21h, 06h функциясы. Тура енгізу-шығару.**

Символдар кодаларын стандартты енгізу құрылғысынан енгізеді немесе стандартты шығару құрылғысына шығарады. Шығару режимінде ASCII кодалары: 07h - қо”ырау, 0Dh - каретканы қайтару, 0Ah - жолды ауыстыру, басқарушы сияқты қарастырылады және өзіне сәйкес әрекетпен орындалады. 08h кодасы - бір қадам артқа жылжу әрекеті, тек шығару қайта бағытталмаған болса орындалады. Енгізу-шығаруды” қайта бағытталуына рұқсат етіледі. Ке”ейтілген ASCII кодасын оқу үшін функцияны қайта орындауды талап етеді. Символ енгізуі болмаған жағдайда оны күтпейді, ол басқаруды программаға қайтарады.

Шақырғанда: AH=06h,

DL= символды” коды (00h-FEH) шығару кезінде,

DL=FFh (енгізу кезінде).

Қайтарғанда: AL=символ кодасы (енгізу кезінде); егер символ болмаса, онда ZF=1.

**INT 21h, 07h функциясы. Бейнесіз сүзгіленбеген енгізу.**

Стандартты енгізу құрылғысынан оны” бейнелеуінсіз символды енгізу. Символ болмаған жағдайда енгізуді күтеді. Енгізуді” қайта бағытталуына рұқсат етіледі. <Ctrl>/C ө”деуін орындамайды. Ке”ейтілген ASCII кодасын оқу үшін функцияны қайта орындауды талап етеді.

Шақырғанда: AH=07h.

Қайтарғанда: AL=мәліметтер кірісіні” байты.

**INT 21h, 08h функциясы. Символды бейнесіз енгізу.**

Стандартты енгізу құрылғысынан оны” бейнеленуінсіз символды жүргізу. Символ болмаған жағдайда енгізуді күтеді. Енгізуді” қайта бағытталуына рұқсат етіледі. Ке”ейтілген ASCII кодын оқу үшін функцияны қайта орындауды талап етеді. Егер енгізу қайта бағытталған болса, BREAK (BREAK=ON) қосылған жағдайында ғана <Ctrl>/C ө”деуін орындайды.

Шақырғанда: AH=08h.

Қайтарғанда: AL= кіріс деректер байты.

**INT 21h, 09h функциясы. Қатарды шығару.** Символ қатарын стандартты шығару құрылғысына шығарады. Қатар \$ символымен аяқталуы керек. ESC-тізбектемесін қолдануға рұқсат етіледі.

ASCII кодалары: 07h - қо”ырау, 08h - бір қадамға артқа жылжу, 0Dh - каретканы қайтару, 0Ah - жолды ауыстыру басқарушы сияқты қарастырылады және өзіне сәйкес әрекетпен орындалады. <Ctrl>/C ө”деуі пернетақтаға енгізген кезде орындалады.

Шақырғанда: AH=09h,

DS:DX=қатар адресі.

**INT 21h, 0Ah функциясы. Пернетақтадан буферленген енгізу.**

Байт жолын стандартты енгізу құрылғысынан, пайдаланушы буферіне стандарттық шығару құрылғысына бейнелеумен енгізеді. Жол (0Dh) каретка қайтару символымен аяқталуы керек. Енгізуді” қайта бағытталуына рұқсат етіледі. Егер енгізу бағытталмаған болса, онда <Ctrl>/C ө”деуін орындайды. Егер енгізу бағытталған болса, онда BREAK (BREAK=ON) қосылған кезінде <Ctrl>/C ө”деуін орындайды.

Шақырғанда: AH=0Ah,

DS:DX=буфер мекені.

Қайтарғанда: Мәліметтер буферге орналасқан.

*Буфер форматы:*

0 байт-күткен қатар ұзындығы.

1 байт-енгізілген қатарды” нақты ұзындығы.

2 байт және әрі қарай- 0DH-пен аяқталатын жол.

**INT 21h, 0Bh функциясы. Енгізу жағдайын тексеру.** Стандартты енгізу алдында буферде символды” бар болуын тексереді. Енгізуді” қайта бағытталуына рұқсат етіледі, егер енгізу бағытталмаған болса, BREAK (BREAK=ON) қосылған кезінде <Ctrl>/C ө”деуін орындайды.

Шақырғанда: AH= 0Bh.

Қайтарғанда: AL= 00h егер символ күтпесе,

AL= FFh егер символ күтіп тұрса.

**INT 21h, 0Ch функциясы. Кіріс буферін тазалау және енгізу.** Пернетақтаны” сақина буферін тазалайды және енгізу функциясын жандандырады. Енгізуді” қайта бағытталуына рұқсат етеді.

Шақырғанда: AH = 0Ch,

AL = талап етілген енгізу функциясыны” нөмірі.

Рұқсат функциялары 01, 07, 08, 0Ah.

DS:DX= буфер адресі (егер AL= 0Ah).

Қайтарғанда: AL=кіріс мәліметтер байты (егер шақыру кезінде AL= 0Ah болса, мәліметтер буферге ығыстырылады).

**INT 21h, 0Eh функциясы. Дискіні та”дау.** Ағымдағы дискіні тағайындайды және жүйедегі логикалық дискілер санын қайтарады.

Шақырғанда: AH= 0Eh.

AL= диск кодасы (0=A:, 1=B: және тағы сол сияқты).

Қайтарғанда: AL= жүйедегі логикалық дискілер саны.

**INT 21h, 1Ah функциясы. Дискімен алмасу аймағыны” адресін орнату.**

Файлдарды басқару блоктары мен операциялары үшін (DTA) дискімен алмасу аймағыны” адресін анықтауға рұқсат етеді.

Шақырғанда: AH= 1Ah.

DS:DX= DTA адресі.

**INT 21h, 25h функциясы. Үзу векторын орнату.**

Үзу векторын үзулерді ө”деу программасыны” адресімен толтыруға рұқсат етеді.

Шақырғанда: AH= 25h,

AL= үзу векторыны” нөмірі,

DS:DX = үзу ө”деуіш адресі.

**INT 21h, 2Ah функциясы. Күнін алу.** Ағымдағы күнін алуға рұқсат етеді.

Шақырғанда: AH=2Ah.

Қайтарғанда: CX=жыл (1980-нен 2099-ға дейін),

DH=ай (1-ден 12-ге дейін),

DL=күн (1-ден 31-ге дейін),

AL=апта күні (0-жексенбі, 1-дүйсенбі, 2-бейсенбі және т.с.с.).

**INT 21h, 2Ch функциясы. Уақытты алу.**

Ағымдағы уақыт мағынасын алу<sup>1</sup>а рұқсат етеді.

Шақырғанда: AH= 2Ch.

Қайтарғанда: CH= сағаттар (0-ден 23-ке дейін),  
CL= минуттар (0-ден 59-ға дейін),  
DH= секундтар (0-ден 59-ға дейін).

**INT 21h, функция 2Fh. Дискімен алмасу аймағыны" адресін алу.**

(DTA) дискімен алмасу аймағыны" адресін қайтарады.

Шақырғанда: AH=2Fh.

Қайтарғанда: ES:DX=DTA адресі.

**INT 21h, 31h функциясы. Программаны" аяқталуы және оны резидентті түрде жадыда сақтау.**

Белсенді программаны" орындалуын аяқтайды және басқаруды аталық процеске қайтарады. DOS-қа қайтару кодасын жібереді. Аяқтау процесі кезінде дискіге буферді жазып тастайды, дескрипторларды жабады, PSP ұяшықтарынан INT 22h, INT 23h, INT 24h векторларын қалпына келтіреді.

Шақырғанда: AH= 31h,

AL= қайтару кодасы,

DX= параграфтардағы қосалқыланған жады көлемі.

**INT 21h, 35h функциясы. Ұзу векторын алу.**

Көрсетілген ұзу векторыны" мазмұнын қайтарады.

Шақырғанда: AH=35h,

AL=ұзу векторыны" нөмірі.

Қайтарғанда: ES:BX= үзулерді ө"деу программасыны" адресі.

**INT 21h, 36h функциясы. Диск жайында ақпарат алу.** Тасушы көлемін және бос ке"істік көлемін анықтауға болатын дискжетек жайында ақпаратты қайтарады. Жоғалтылған кластерлер бос емес деп саналады.

Шақырғанда: AH=36h,

DL= дисковод кодасы (0= ағымды, 1=A: және т.с.с.).

Қайтарғанда: AX=кластердегі секторлар саны,

BX= бос кластерлер саны,

CX= байттағы секторлар өлшемі,

DX= дискдегі толық кластерлер саны,

Қайтарғанда қате кезінде: AX= FFFFh.

**INT 21h, 39h функциясы. Каталогты құру.**

Каталогты көрсетілген жолды" аяғында жасайды.

Шақырғанда: AH= 39h,

DS:DX= ASCIIZ жолы түріндегі каталогты" адресі.

Қайтарғанда қате кезінде: CF= 1,

AX= қате кодасы.

**INT 21h, 3Ah функциясы. Каталогты жою.**

Көрсетілген каталогты жояды.

Шақырғанда: AH= 3Ah,

DS:DX= ASCIIZ жолы түріндегі каталог адресі.

Қайтарғанда қате кезінде: CF= 1,

AX= қате кодасы.

**INT 21h, 3Bh функциясы. Ағымдағы каталогты орнату.**

Жа”а ағымдағы каталогты орнатады.

Шақырғанда: AH= 3Bh,

DS:DX= ASCIIZ жолы түріндегі каталогтар адресі.

Қайтарғанда қате кезінде: CF= 1,

AX= қате кодасы.

**INT 21h, 3Ch функциясы. Файлды құру.** Берілген спецификациямен жа”а файлды құрады. Егер көрсетілген файл бар болса, ол нөлдік ұзындыққа дейін қиылады. Кез келген жағдайда файл ашылады және оны” дескрипторы кейінгі операциялар үшін қайтарылады.

Шақырғанда: AH= 3Ch,

CX= файл атрибуттары (құралуы мүмкін):

1 - тек оқу үшін,

2 – жасырын,

4 - жүйелік,

8- том белгісі,

20h - архив атрибуты.

DS:DX= ASCIIZ жолы түріндегі файл спецификациясыны” адресі.

Қайтарғанда: AX= дескриптор.

Қате кезінде: CF= 1,

AX= қате кодасы.

**INT 21h, 3Dh функциясы. Файлды ашу.**

Берілген спецификациямен файлды ашады. Келесі операциялар үшін дескрипторды қайтарады. Көрсеткішті файлды” басына (байт 0) апарып орналастырады.

Шақырғанда: AH= 3Dh.

AL= рұқсат әлпісі:

0 - оқу,

1 – жазу,

2 - оқу және жазу.

*Ескерту:* егер режимге 80h қосылған болса, туынды процесс дескрипторға ие болады. Кері жағдайда туынды процесс дескрипторға ие болмайды.

DS:DX = ASCIIZ жолы түріндегі файл спецификациясының адресі.

Қайтарғанда: AX= дескриптор

Қате кезінде: CF= 1,

AX= қате коды.

#### **INT 21h, 3Eh функциясы. Файлды жабу.**

Файлдарды" ішкі буферлерін дискіге жазып тастайды, файлды жабады және дескрипторды босатады. Егер файл түрін өзгерткен болса, каталог жазбасында файлды құратын жа"а мағыналы дата және уақыт орналасады.

Шақырғанда: AH=3Eh,

BX= дескриптор.

Қайтарғанда қате кезінде: CF= 1,

AX= қате кодасы.

#### **INT 21h, 3Fh функция. Файлдан немесе құрылғыдан оқу.**

Файлдан деректерді қолданушы буферіне жібереді және нұсқағышты өзгертеді. ASCII түрінде символдық құрылғыдан оқу кезінде көрсетілген жол ұзындығы оқылады немесе ұзындығы со"ына келмей "Enter" пернесі басылғанша символдар енгізіледі.

Шақырғанда: AH= 3Fh,

BX= дескриптор,

CX= байт саны,

DS:DX= қолданушы буфер мекені.

Қайтарғанда: AX= оқылған байт саны.

Қате кезінде: CF= 1,

AX= қате кодасы.

#### **INT 21h, 40h функциясы. Файлға немесе құрылғыға жазу.**

Файлға деректерді қолданушы буферіне жібереді және нұсқағышты өзгертеді. Егер шақыру кезінде CX= 0 болса, онда файл ұзындығы сәйкесінше ағымды нұсқағыш жағдайында көрсетіледі. Шақырғанда: AH= 40h,

BX= дескриптор,

CX= байт саны,

DS:DX= қолданушы буфер мекені.

Қайтарғанда: AX= жіберілген байт саны.

Қате кезінде: CF= 1,

AX= қате кодасы.

**INT 21h, 41h функциясы. Файлды жою.**

Белгіленген файлды (тек біреуін) жояды.

Шақырғанда: АН= 41h,

DS:DX= ASCIIZ жол түріндегі файлды” спецификациясы.

Қайтарғанда қате кезінде: CF= 1,

AX= қателік кодасы.

**INT 21h, функция 42h. Файл ішінде нұсқағышты орналастыру.**

Нұсқағышты” ағымды орнын файлды” кез-келген байтына (келесі операцияларды” файлға тікелей қатысуымен орындау үшін) орнатуға рұқсат береді.

Шақырғанда: АН = 42h.

AL = нұсқағышты орнату т,сілі:

0- файлды” басынан ығыстыру,

1- ағымды орнынан ығыстыру,

2- файлды” со”ынан ығыстыру.

BX= дескриптор.

CX= ығыстыруды” үлкен жартысы,

DX= ығыстыруды” кіші жартысы.

Қайтарғанда: DX= “айтарыл”ан нұсқағышты” үлкен жартысы,

AX= “айтарыл”ан нұсқағышты” кіші жартысы.

**INT 21h, 43h функциясы. Файл атрибутын алу немесе орнату.**

Файл немесе каталог атрибутыны” мағынасын өзгертуге немесе алуға рұқсат етеді. Файлды каталогқа немесе белгі томына қайта ө”деуге болмайды. Каталогты “жасырын” түрінде жасауға болады.

Шақырғанда: АН= 43h.

AL= 0 атрибуттарды алу үшін.

AL= 1 атрибуттарды орнату үшін.

CX= файл атрибуты (комбинациялануы мүмкін):

1- тек оқу үшін,

2- жасырын,

4- жүйелік,

20h- архивтеу атрибуты.

DS:DX= каталог немесе файл мекені.

Қайтарғанда: CX= қайтарылған файл атрибуты (егер шақыру кезінде AL= 0 болса).

Қате кезінде: CF= 1,

AX= қате кодасы.

**INT 21h, 47h функциясы. Ағымдағы каталогты оқу.**

Ағымдағы каталогқа, оны” атын қоса, ASCIIZ жолын толық жолмен қайтарады (түпкі каталогтан). Ағымдағы диск және түпкі каталог белгісін қайтармайды.

Шақырғанда: АН=47h,  
DL= диск кодасы (0=ағымды , 1=A, 2=B, ж.с.с.),  
DS:SI=64-байт өлшемді аралық жады адресі.

Қайтарғанда: Аралық жады ағымды каталог спецификациясымен толтырылған.

Қате кезінде: CF=1,  
AX=қате кодасы.

**INT 21h, 48h функциясы. Жады блогын бөліп беру.**

Жады блогын бөліп береді және оны” сегменттік адресін қайтарады.

Шақырғанда: АН= 48h,  
BX=оны” жады талап етілген параграфтарыны” саны.  
Қайтарғанда: CF=1, AX=қате кодасы.  
BX=параграфпен берілген бос блогыны” е” үлкен өлшемі.

**INT 21h, 49h функциясы. Жады блогын босату.**

Жады блогын босатады және оны басқа программалармен қолдану үшін жүйеге береді.

Шақырғанда: АН=49h,  
ES=босатылатын блокты” сегменттік адресі.  
Қате кезінде: CF=1,  
AX= қате кодасы.

**INT 21h, 4Ah функциясы. Жадыны” бөліп алынған блогыны” өлшемін өзгерту.**

Жадыны” бөліп алынған блог өлшемін ұлғайтады немесе кемітеді .

Шақырғанда: АН=4Ah,  
BX=параграф бойынша талап етілген блог өлшемі.  
ES=көлемі өзгерілетін блокты” сегменттік адресі.  
Қайтарғанда:  
BX=параграфпен берілген бос блогыны” е” үлкен өлшемі.  
Қате кезінде: CF=1, AX=қате кодасы.

**INT 21h, 4Bh функциясы. Программаларды орындауға жіберу (Ehес функциясы).**

Қолданбалы программаға (аталық процеске) басқа программаны (туынды процесті) жүктеуге және орындауға

мүмкіндік береді. Орындауға жіберілген программа аяқталғаннан кейін, басқару аталық процеске қайтып оралады.

Шақырғанда: АН=4Bh.

AL - ішкі функцияны” нөмірі:

AL=0 программаны жүктеу және орындау,

AL=1 программаны жүктеу, бірақ орындамау,

AL=2 программаны жүктеу (құжатталмаған ішкі функция),

AL=3 программаны оверлей ретінде жүктеу (PSP блогын құрып алмай),

AL=5 программаны орындау үшін дайындау.

ES:BX= параметрлер блогыны” адресі.

DS:DX=туынды программаға жолды көрсететін ASCIIZ қатар.

Қате кезінде : CF=1,

AX=қате кодасы.

**INT 21h, 4Ch функциясы. Процесті” аяқталуы.**

Аяқтау кодасын оған арналған DOS-ты” (SDA) ағымдағы деректер аймағыны” байтына енгізіп ағымдағы процесті аяқтайды. Осы кезде функция процеске бөліп берген жадыны босатады, буферлерді дискіге жазып тастайды, барлық ашық дескрипторларды жабады, PSP ұяшығынан INT 22h, INT 23h, INT 24h векторларын қалпына келтіреді.

Шақырғанда: АН=4Ch,

AL=аяқтау кодасы.

**INT 21h, 4Dh функциясы. Аяқтау кодасын алу.**

EXEC функциясымен (INT 21h, 4Bh функциясы) жандандырылған туынды процесінен қайтқаннан кейін аталық процесс қолданылады.

Шақырғанда: АН= 4Dh.

Қайтарғанда: АН= аяқталу түрі:

00h – INT 20h немесе INT 21h (00h немесе 4Ch функциясы) көмегімен қалыпты аяқталу,

01h – пайдаланушы < Ctrl>/ C енгізді,

02h- критикалық қатені” драйвері арқылы аяқталуы.

AL=туынды процесінен айтарылған кодасы.

**INT 21h, 4Eh функциясы. Бірінші файлды табу.**

Топты операцияларыны” алыбы (шаблон) бойынша көрсетілген каталогтағы бірінші файлды іздейді. Егер CX=8, тек том белгісі іздестіріледі. Басқа атрибуттар немесе оларды”

комбинациялары көрсетілсе, көрсетілген атрибуттары бар және жай файлдар іздестіріледі.

Шақырғанда: AH=4Eh,

CX=іздестіріліп жатқан файл атрибуты (комбинациялана алады):

1 – тек оқу үшін, 2-жасырын, 4-жүйелік, 8-том белгісі, 10h – каталог, 20h – архивтеу атрибуты.

DS:DX= іздестіріліп жатқан файлды" спецификациясыны" адресі.

Қайтарғанда: файл аты және ке"ейтуі, дискімен алмасу аймақты" 1Eh...2Ah байттына енгізіледі.

Қате кезінде: CF=1, AX=қате кодасы.

**INT 21h, 4Fh функциясы. Келесі файлды табу.**

Келесі файлды іздеу 4Eh функциясы бірінші файлды тапқаннан кейін іске асырады. 4Eh функциясы сәтті орындалғаннан кейін ғана қолданылады. Егер "алып"а сәйкес файлдар екіден көп болса, функцияны CF=1 ("алып"а сәйкес файлдар жоқ) алғанша бірнеше рет орындауға тура келеді .

Шақырғанда: AH=4Fh.

Қайтарғанда: 4Eh функциясын қара.

Қате кезінде: CF=1, AX=қате кодасы.

**INT 21h, 52h функциясы (құжатталмаған). Байланыс векторлық кестесіні" адресін алу.**

Шақырғанда: AX=5200h.

Қайтарғанда: ES:BX=кесте адресі.

**INT 21h, 56h функциясы. Файлды" атын өзгерту.**

Файлды" атын өзгерту немесе оны осы дискіде басқа каталогқа орналастыру. Каталогты" атын өзгертуге мүмкіндігі бар.

Шақырғанда: AH=56h,

DS:DX= ағымды файл спецификациясыны" мекені,

ES:DI=жа"а файл спецификациясыны" мекені.

Қате кезінде: CF=1,

AX=қате кодасы.

**INT 21h, 57h функциясы. Күні мен уақытты қою немесе алу.**

Каталогта жазылған файл жасалған күні мен уақытты алуға немесе қоюға мүмкіндік береді.

Шақырғанда:

уақыт пен күнді алу үшін:

AH=57h,

AL=0,

ВХ= дескриптор;  
уақыт пен күнді қою үшін:

АН= 57h,

AL=1,

ВХ= дескриптор.

СХ=уақыт (0...4 биттер – екі секундты аралықтар, 5h...Ah-минуттар, Bh...Fh –сағаттар).

DX=күн (0...4 биттер - күн, 5...8 - ай, 9h...Fh – жыл, 1980 жылға қатысты).

Қайтарғанда: Егер шақыру кезінде AL= 0:

СХ= уақыт,

DX= күн.

Қате кезінде: CF=1,

АХ=қате кодасы.

**INT 21h, 5Ah функциясы. Уақытша файлды құру.**

Көрсетілген каталогта, көрсетілген атрибуттармен файлды жасайды және дескриптормен файл атын қайтарады.

Шақырғанда: АН=5Ah.

СХ= файл атрибуты:

1-тек оқу үшін,

2- жасырын,

4 – жүйелік,

20h-архив атрибуты.

DS:DX=ASCIIZ жолы түріндегі файлды” толық спецификациясыны” адресі.

Қате кезінде: CF=1, АХ=қате кодасы.

**INT 21h, 5Bh функциясы. Жа”а файлды құру.**

Көрсетілген спецификациямен және атрибуттармен жа”а файлды жасайды және дескрипторды қайтарады. Каталогта осындай файл бар болса, функция атемен аяталады.

Шақырғанда: АН=5Bh.

СХ= файл атрибуты:

1-тек оқу үшін,

2-жасырын,

4-жүйелік,

8-том белгісі.

DS:DX=ASCIIZ жолы түріндегі файл спецификациясыны” адресі.

Қайтарғанда: АХ =дескриптор.

Қате кезінде: CF= 1,

АХ=қате кодасы.

**INT 25h. Дискіден абсолютты түрде оқу.**

Жадыға дискіден бір немесе сектор тобы берілген, бастапқы салыстырмалы нөмірімен оқуға мүмкіндік береді. Секторлар 0-ден бастап, логикалық (физикалық емес!) дискіден бастап нөмірленеді. Осылайша берілген логикалық дискіні” жүктемелік секторыны” 0 нөмірі, кейін дискіде FAT-ты” бірінші көшірмесіні” бірінші секторы – 1 нөмірі болады.

INT 25h және INT 26h үзулерін орындағаннан кейін тапсырма стегінде жалауша регистріні” мәнін сақтайтын сөз қалады. Егер бұл сөзді жоймаса, онда программаны” келесі қадамдары бұзылады.

Шақырғанда: AL=диск нөмірі ( 0= A, 1= B және т.б.),

CX= оқылатын сектор саны,

DX= бірінші оқылатын секторды” салыстырмалы нөмірі,

DS:BX= буферді” адресі.

Қате кезінде: CF= 1,

АХ=қате кодасы.

Ескерту:

Функция қате кодасын АН және AL регистрінде қайтарады.

АН регистріндегі қате кодасы:

01h- дұрыс емес команда,

02h- дұрыс емес адрес белгісі,

04h- сұратыл'ан секторы табылмаған,

08h- жадыға тікелей қатынауды” қатесі,

10h- деректер қатесі (дұрыс емес бақылау қосындысы),

20h-контроллер қатесі,

40h- позициялану қатесі.

AL регистріндегі қате кодасы:

00h-жазбаны қорғау қатесі,

01h- белгісіз құрылғы,

02h- дискжетек дайын емес,

03h- белгісіз команда,

04h- деректер қатесі (дұрыс емес бақылау қосындысы),

06h- позициялану қатесі,

07h- тасушыны” белгісіз түрі,

08h- сектор табылған жоқ.

**INT 26h. Дискіге абсолютты жазу.**

Жадыдан дискіге бір немесе сектор тобын, бастапқы берілген салыстырмалы нөмірмен жазуға мүмкіндік береді. Секторлар 0 –

ден логикалық (физикалық емес) дискіден бастап нөмірленеді. Осылайша берілген логикалық дискіні” жүктемелі сектор 0 нөмірі, содан кейін дискіде FAT-ты” бірінші көшірмесіні” бірінші секторы – 1 нөмірі болады.

INT 25h және INT 26h үзулерін орындағаннан кейін, тапсырма стегінде жалауша дискіні” мәнін сақтайтын сөз қалады. Егер бұл сөз жойылса программаны” келесі қадамдары бұзылады.

Шақырғанда: AL=дискжетек нөмірі ( 0=A, 1=B және т.б.),

CX= оқылатын секторлар саны,

DX= бірінші оқылатын секторды” салыстырмалы нөмірі,

DS:DX= буфер адресі.

Қате кезінде: CF =1,

AX= қате кодасы.

Ескертуді INT 25h – тан қара.

### **INT 2Fh. Мультиплексті үзу.**

Үзу процестері арасындағы байланысты ұйымдастыру үшін және жүйелік пен қолданбалы резидентті программаларымен ақпаратты алмастыру<sup>1</sup>а арналған. Пайдаланушылар<sup>1</sup>а C0h...FFh функциялары арнал<sup>1</sup>ан.

Шақырғанда: AH=функция,

AL= ішкі функция.

Басқа регистрлер қажеттілігі бойынша қолданылады.

Қайтарғанда: AL=0 егер программа орнатылмаған болса, оны орнатуға болады.

AL=1 егер программа орнатылмаған болса, оны ,рі орнатуға болмайды.

AL=FFh егер программа орнатылған болса.

Қате кезінде: CF= 1,

AX=қате кодасы.

## **Б КОСЫМШАСЫ BIOS ФУНКЦИЯЛАРЫ БОЙЫНША АНЫҚТАМАЛЫҚ МӘЛІМЕТТЕР**

### **INT 10h, 00h функциясы. Бейнережимді орнату.**

Бұл функция ағымдағы бейнережимді орнатады.

*Функцияны шақырғанда:* АН = 00h.

AL = бейнережим:

03h - мәтіндік, 80\*25 қатар, 16 түс,

10h – графиктік, 640\*350 нүктелер, 16 түс,

04h - графиктік, 320\*200 нүктелер, 4 түс (CGA бейнеадаптері).

### **INT10h, 01h функциясы. Ме”зер (курсор) өлшемін орнату.**

Мәтіндік бейнережимдерде жыпылықтайтын аппараттық ме”зер жаймалауды” (развертка) бастапқы және со”ғы қатарларын орнатуға мүмкіндік береді.

*Функцияны шақырғанда:* АН = 01h,

CH 0...4 биттері = бастапқы қатар,

CL 0...4 биттері = со”ғы қатар.

### **INT 10h, 02h функциясы. Ме”зер орнын орнату.**

Көрсетілген мәтіндік бетті” (соны” ішінде белсенді емес бетте де), мәтіндік координаттарындағы ме”зер орнын көрсетеді.

*Функцияны шақырғанда:* АН = 02h,

BH = бет,

DH = қатар,

DL = баған.

### **INT 10h, 03h функциясы. Ме”зерді” ағымдағы орнын анықтау.**

Берілген беттегі (соны” ішінде белсенді емес бетте де) ме”зерді” орнын қайтарады.

*Функцияны шақырғанда:* АН = 03h,

BH = бет.

*Функциядан қайтарғанда:*

CH = ме”зер жаймасыны” бастапқы қатары,

CL = ме”зер жаймасыны” со”ғы қатары,

DH = қатар,

DL = баған.

### **INT 10h, 05h функциясы. Бейнебетті орнату.**

Белсенді түрде бейнебетті орнатады (мәтіндікті де, графикті де). *Функцияны шақырғанда:* АН = 05h,

AL = бет.

**INT 10h, 06h функциясы. Бастапқы терезені жасау немесе терезені жоғары айналдыру (жылжыту).**

Көрсетілген координаттар бойынша берілген атрибуттарымен бос орындарымен (ASCII кодасы - 20h) бастапқы терезені жасайды немесе терезе мазмұнын берілген қатар санына жоғары қарай айналдырады (тек белсенді терезеге арналған). Айналдыру кезінде төменде пайда болатын қатарлар ASCII бос орындарымен (берілген атрибут бойынша) толтырылады. Бұл функцияны экранды тез тазалау үшін немесе кез-келген экрандағы төртбұрышты аймақты тазалауға пайдалануға болады.

*Функцияны шақырғанда:* АН = 06h,

AL = айналдыру қатарлар саны, егер AL = 0 болса, онда терезені” бүкіл бойы тазаланады,

BH = терезедегі символдар атрибуты,

CH = терезені” сол жақ жоғарғы бұрышыны” Y-координатасы,

CL = терезені” сол жақ жоғарғы бұрышыны” X-координатасы,

DH = терезені” о” жақ төменгі бұрышыны” Y-координатасы,

DL = терезені” о” жақ төменгі бұрышыны” X-координатасы.

**INT 10h, 07h функциясы. Бастапқы терезені жасау немесе терезені төмен айналдыру (жылжыту).**

Көрсетілген координаттар бойынша берілген атрибуттарымен ASCII бос орындарымен бастапқы терезені жасайды немесе терезе мазмұнын берілген қатар санына төмен қарай айналдырады (тек белсенді терезеге арналған). Айналдыру кезінде жоғарыда пайда болатын қатарлар ASCII бос орындарымен (берілген атрибут бойынша) толтырылады. Бұл функцияны экранды тез тазалау үшін немесе кез-келген экрандағы төртбұрышты аймақты тазалауға пайдалануға болады.

*Функцияны шақырғанда:* АН = 07h,

AL = айналдыру қатарлар саны, егер AL = 0 болса, онда терезені” бүкіл бойы тазаланады,

BH = терезедегі символдар атрибуты,

CH = терезені” сол жақ жоғарғы бұрышыны” Y-координатасы,

CL = терезені” сол жақ жоғарғы бұрышыны” X-координатасы,

DH = терезені" о" жақ төменгі бұрышыны" Y-координатасы,  
DL = терезені" о" жақ төменгі бұрышыны" X-координатасы.

**INT 10h, 08h функциясы. Ме"зерді" ағымды орнындағы символды және атрибутты оқу.**

Берілген беттегі ме"зерді" ағымдағы орнын ASCII – символын және оны" атрибутын қайтарады (белсенді емес бетке де қатысты).

*Функцияны шақырғанда:* AH = 08h,

BH = бет.

*Функциядан қайтарғанда:*

AH = атрибут,

AL = символ.

**INT 10h, 09h функциясы. Ме"зерді" ағымдағы орнына символды және атрибутты жазу.**

Ме"зерді" ағымдағы орнына символды және атрибутты графиктік пен мәтіндік режимде жазады. Графикалық режимде символдар келесі қатарға өтпеуі қажет, AL-дағы барлық кодалар та"ба ретінде қаралып, ме"зерді" орнын басқармайды. Шығару операциясы аяқталған со", ме"зерді келесі орнына 02h функциясы көмегімен жылжытады. Қайталау коэффициенті көмегімен біркелкі символдар қатарларын шығаруға болады (бірақ ме"зер ығыспайды). Мәтіндік режимде символ берілген атрибутпен шығарылады (берілген негізгі түсте символ белгілі бір түспен беріледі). Графикалық режимде BL мазмұны тек символ түсіне ғана әсер етеді, ал фон түсіне әсері жоқ, бірақ ол кезде та"ба орындағы бейнеленген графикалық бейне өшіріліп қалады.

*Функцияны шақырғанда:* AH = 09h,

AL = символ,

BH = бет,

BL = атрибут (мәтіндік режим) немесе түс (графиктік режим),

CX = қайталау коэффициенті.

**INT 10h, 0Ah функциясы. Символды ағымдағы ме"зер орнына жазу.**

Ме"зерді" ағымды орнындағы ASCII символын графиктік және мәтіндік режимде жазады. Символ осы орынға алдын-ала тағайындалған атрибутты қабылдайды. Графикалық режимде символдар келесі қатарға өтпеуі қажет, AL-дағы барлық кодалар та"ба ретінде қаралып, ме"зерді" орнын басқармайды. Шығару операциясы аяқталған со", ме"зерді келесі орнына 02h функциясы көмегімен жылжытады. Қайталау коэффициенті

көмегімен біркелкі символдар қатарларын шығаруға болады (бірақ ме”зер ығыспайды).

*Функцияны шақырғанда:* АН = 0Ah,

AL = символ,

BH = бет,

CX = қайталау коэффициенті.

**INT 10h, 0Ch функциясы. Пиксельді жазу.**

Бейне буферге берілген графикалық позициядағы берілген түспен нүктені жазады.

*Функцияны шақырғанда:* АН = 0Ch,

AL = түс,

BH = бет,

CX = графикалық баған,

DX = графикалық қатар.

**INT 10h, 0Eh функциясы. Телетайп режимінде символды жазу.**

Белсенді беттегі, ағымдағы ме”зер орнына ASCII символын жазады және ме”зерді келесі позицияға ығыстырады. ASCII кодалары: 07h-қо”ырау, 08h-артқа бір қадам, 0Dh-қаретканы қайтару, 0Ah-қатарды ауыстыру - бұл басқару ретінде қарастырылып, соған сәйкес амалдар орындалады. Қалған басқару кодалары та”ба ретінде қарастырылып экранға шығарылады. Ме”зерді келесі қатарға автоматты түрде ауыстыру және экран скроллингін (экран символдармен толтырылғаннан кейін экранны” мазмұны бір қатарға жоғарыға жылжытылып тұрады) іске қосу. Символ атрибутын көрсетуге болмайды. Жазба кезінде, ағымдағы орынға орнатылған атрибут орындалады.

*Функцияны шақырғанда:* АН = 0Eh,

AL = символ,

BL = символ түсі (графикалық режим),

BH = бет.

**INT 10h, 0Fh функциясы. Бейнережимді алу.**

Бейнеконтроллерді” ағымдағы бейнережимін алуға мүмкіндік береді.

*Функцияны шақырғанда:* АН = 0Fh.

*Функциядан қайтарғанда:*

АН = экрандағы символдар бағаныны” саны,

AL = бейнережим,

BH = белсенді бейнебет.

INT 10h, 10h функциясы, 00h ба”ыны”сы функциясы. Tscтік регистрді баптау.

Регистр түсіні" нөмірі пиксель түсіні" нөміріне сәйкес орналастырылады.

*Функцияны шақырғанда:* AH = 10h, AL=00h.

BH = ҚЖК (Қызыл Жасыл Көк) кодасындағы түс мәні ,

BL = түстік регистр (0...Fh).

**INT 10h, 10h функциясы, 01h бағыны"қы функциясы. Экранға шетіні" түсін орнату.**

Экран шетіні" түсін орнатады (жаймалау шетін).

*Функцияны шақырғанда:* AH = 10h, AL=01h,

BH = ҚЖК кодасындағы түс мәні,

BL = түстік регистр (0...Fh).

**INT 10h, 10h функциясы, 02h бағыны"қы функциясы. Түстік палитраны баптау және экранға шетіні" түсін орнату.**

Түстік регистрлер нөмірі пиксель түсіні" нөміріне сәйкес орналастырылады және экранға шетіні" түсін орнатады (жаймалау шетін).

*Функцияны шақырғанда:* AH = 10h, AL=02h,

ES:DX = түс кестесіні" адресі, ұзындығы 17 байт. Кесте қжкҚЖК кодаларымен толтырылады. Олар 0...15 (бірінші 16 байт) түстік регистрлеріне және экран шетіні" регистріне (со"ғы байт) жүктеледі.

**INT 10h, 10h функциясы, 03h бағыны"қы функциясы. "Жыпылықтату/жарықтық" (мерцание/яркость) битін ауыстырып қосу.**

Символ атрибутыны" үлкен битті (7) мақсатын анықтайды: символды" жыпылықтауын немесе фон түсіні" қоюлығын көбейту.

*Функцияны шақырғанда:* AH = 10h, AL=03h,

BL = атрибутты" үлкен биттін тағайындау: 0-атрибут жарықтығыны" (қоюлығыны") е" көбі, 1-символды" жыпылықтауы.

**INT 10h, 10h функциясы, 07h бағыны"қы функциясы. Түстік регистрді оқу.**

Түстік регистрді" мазмұнын қайтарады.

*Функцияны шақырғанда:* AH = 10h, AL=07h.

*Функциядан қайтарғанда:*

BH = қжкҚЖК кодасындағы түс мәні.

**INT 10h, 10h функциясы, 09h бағыны"қы функциясы. Түстік палитраны және экран шетіні" түсін оқу.**

Барлық 17 түстік регистрлерді" мазмұнын бір командамен алуға мүмкіндік береді.

*Функцияны шақырғанда:* АН = 10h, AL=09h,

ES:DX = түстік регистрлерді” мазмұндары үшін 17-байттық буфері.

**INT 10h, 11h функциясы, 03h бағыны”қы функциясы. Та”ба генераторы блогыны” спецификаторын тағайындау.**

Мәтіндік бейнережимдегі символдар байт атрибутыны” үшінші битіні” функциясы қызметін анықтауға мүмкіндік береді. Бұл функция қызметіне: символ жарықтығын немесе та”ба генераторыны” блок нөмірін жатқызуға болады.

*Функцияны шақырғанда:* АН = 11h, AL=03h,

BL = та”ба генераторы блогыны” кодасы.

**Ескерту:**

BL регистріні” 0...1 биттері та”ба генераторыны” блок нөмірін (0...3) анықтайды. Егер атрибут байтыны” 3-ші биті 0-ге те” болса, онда та”ба генератордан символдар экранға түсе бастайды.

Егер атрибут байтыны” 3-ші биті 1-ге те” болса, онда та”ба генератордан символдар экранға түсе бастайды. Бұл жағдайда BL регистріні” 2...3 биттері та”багенераторыны” блок нөмірін (0...3) анықтайды.

Егер 0...1 және 2...3 биттер мәндері дәлме-дәл келсе, онда та”ба генераторыны” бір ғана блогы пайдаланылады (256 символ), ал атрибуттар байтыны” 3-ші биті символдар жарықтығын басқарады.

INT 10h, 11h функциясы, 10h бағыны”қы функциясы. Пайдаланушы еарпін жсктеу.

Пайдаланушыны” белгіленген қаріптері бар кестесін, көрсетілген символдар генератор блогына жүктейді. Контроллерді қаріпті” жа”а өлшеміне қайтадан программалайды. 10h бағыны”қы функциясын пайдаланғанда 0 бейнебет белсенді болу керек, 00h бағыны”қы функциясы да осындай қызмет атқарады, бірақ ол бейнеконтроллерді қайтадан программаламайды. Бұл бағыны”қы функцияны пайдаланғанда кез келген бейнебет белсенді бола алады.

*Функцияны шақырғанда:* АН = 11h, AL=10h,

BH = графикалық нүктелердегі символ биіктігі,

BL = генератор блогы,

CX = кестеде сипатталған символдар саны,

DX = кестені” 1-ші символына арналған код,

ES:BP = кесте адресі.

**INT 10h, 11h функциясы, 11h бағыны"қы функциясы. ТСҚ (ПЗУ) 8x14 қарпін жүктеу.**

8x14 өлшемді ТСҚ (ПЗУ) BIOS графикалық нүктелер қарпін жүктейді. Егер үнсіз келісім бойынша болса, онда символдар генераторыны" көрсетілген блогына жүктеледі. 11h бағыны"қы функциясын пайдаланғанда 0 бейнебеті белсенді болуы керек. 01h бағыны"қы функциясыны" да атқаратын қызметі осындай, бірақ контроллерді қайтадан программаламайды. Бұл бағыны"қы функцияны пайдаланғанда кез келген бейнебет белсенді бола алады.

*Функцияны шақырғанда:* AH = 11h, AL=11h,  
BL = генератор блогы.

**INT10h, 11h функциясы, 12h бағыны"қы функциясы. ТСҚ 8x8 қарпін жүктеу.**

8x8 өлшемді ТСҚ (ПЗУ) BIOS графикалық нүктелер қарпін жүктейді. Егер үнсіз келісім бойынша болса, онда символдар генераторыны" көрсетілген блогына жүктеледі. 11h бағыны"қы функциясын пайдаланғанда 0 бейнебеті белсенді болуы керек. 02h бағыны"қы функциясыны" да атқаратын қызметі осындай, бірақ контроллерді қайтадан программаламайды. Бұл бағыны"қы функцияны пайдаланғанда кез келген бейнебет белсенді бола алады.

*Функцияны шақырғанда:* AH = 11h, AL=12h,  
BL = генератор блогы.

INT10h, 11h функциясы, 21h бағыны"қы функциясы. 43h векторын пайдаланушы сарпіне орнату.

43h векторына, графикалық режимде қолдану үшін, пайдаланушы қарпі кестесіні" адресін жүктейді. Дәл сол мезгілде BIOS бейнедеректерді" аймағы да өзгертіледі.

*Функцияны шақырғанда:* AH = 11h, AL=21h,  
BL = экрандағы қатар саныны" кодасы:

00h - пайдаланушы көрсетеді (DL регистрін қара),

01h- 14 қатар,

02h- 25 қатар,

03h- 43 қатар.

CX = бір символдағы пиксель (байт) қатарыны" саны,

DL = экрандағы қатар саны (егер BL = 00h),

ES:BP = пайдаланушы қарпі кестесіні" адресі.

INT 10h, 11h функциясы, 22h бағыны"қы функциясы. 43h векторын TCI 8x14 сарпіне орнату.

ТСК BIOS-тан 43h векторына, графикалық режимде қолдану үшін 8x14 өлшемді матрицалы қарпі кестесіні” адресін жүктейді. Дәл сол мезгілде BIOS бейнедеректерді” аймағы да өзгертіледі.

*Функцияны шақырганда:* АН = 11h, AL=22h.

INT 10h, 11h функциясы, 23h бағыны”ғы функциясы. 43і векторын TCI 8x8 сарпіне орнату.

ТСК BIOS-тан 43h векторына, графикалық режимде қолдану үшін 8x8 өлшемді матрицалы қарпі кестесіні” адресін жүктейді. Дәл сол мезгілде BIOS бейнедеректерді” аймағы да өзгертіледі.

*Функцияны шақырганда:* АН = 11h, AL=23h.

**INT 10h, 11h функциясы, 30h бағыны”қы функциясы. Қаріптер жайлы мәлімет алу.**

Қаріптер кестесі адрестерін және аталған қаріп символындағы пиксель (байт) қатарлар санын алуға мүмкіндік береді.

*Функцияны шақырганда:* АН = 11h, AL=30h.

BH = қаріп кодасы:

00h - 1Fh векторыны” ағымдағы мазмұны,

01h - 43h векторыны” ағымдағы мазмұны,

02h - ТСК 8x14 қарпі,

03h - ТСК 8x8 қарпі (1-ші жартысы),

04h - ТСК 8x8 қарпі (2-ші жартысы).

*Функциядан қайтарғанда:*

CX = бір символдағы пиксель (байт) қатарыны” саны,

DL = экрандағы со”ғы қатар нөмірі,

ES:BP = қаріп кестесіні” адресі.

**INT 10h, 13h функциясы. Телетайп режимінде қатарды жазу.**

Бейнебуферді” ағымдағы бетіне қатарды, көрсетілген орыннан бастап жазады. ASCII кодалары: 07h - қо”ырау, 08h - артқа бір қадам, 0Dh - қаретканы қайтару, 0Ah - қатарды ауыстыру - бұл басқару ретінде қарастырылып, соған сәйкес амалдар орындалады.

*Функцияны шақырганда:* АН = 13h.

AL = жазу режимі:

0 - атрибут BL-де, қатарда тек символдар кодасы бар, ме”зер жазба орындалған со” ығыспайды (жылжымайды);

1 - атрибут BL-де, қатарда тек символдар кодасы бар, ме”зер жазба орындалған со” ығысады;

2 - қатарда атрибуттарды” және символдарды” кодасы кезектесіп болып тұрады, ме”зер жазба орындалған со” ығыспайды;

3 - қатарда атрибуттарды" және символдарды" кодасы кезектесіп болып тұрады, ме"зер жазба орындалған со" ығысады.

BH = бет.

BL = атрибут (егер AL = 0 немесе 1).

CX = символдық қатар ұзындығы (ұзындық өлшеміне тек символдар кодасы кіреді, бірақ атрибуттар байттары кірмейді).

DH = экрандағы қатарлар нөмірі.

DL = экрандағы бағандар нөмірі.

ES:BP = қатар адресі.

**INT 13h, 00h функциясы. Дискілік жүйені босату.**

Дискілік контроллерді алғашқы күйіне келтіреді. 0 цилиндр позициясына бастиекті орнатады және жүйені енгізу-шығаруға дайындайды.

*Функцияны шақырганда:* AH = 00h.

DL = дискжетек:

00h...07h - иілгіш диск,

80h...FFh - қатты диск.

*Қате болғанда:* CF= 1,

AH = күйі (ескертуді қара).

**Ескерту:** Күйлер кодалары келесі мәндерді қабылдауы мүмкін:

00h — қате болмайді,

01h — дұрыс емес команда,

02h — адрес та"басы табылмаған,

03h — дискетаға жазуға рұқсат жоқ,

04h — сектор табылмаған,

05h — қатты дискіні босату (сброс) іске асқан жоқ,

06h — дискета алынып тұр,

07h — қатты дискіні" параметрлер кестесі дұрыс емес,

0Ch — деректер тасушысыны" типі табылмады,

0Dh — қатты дискі пішімінде (форматында) секторларды" саны дұрыс емес,

10h — деректерді" қалпына келмейтін қатесі,

11h — қатты дискідегі деректерді" қалпына келетін қатесі,

20h — контроллер қатесі,

40h — орын (позиция) қатесі,

80h — диск тайм-ауты,

AAh — қатты диск дайын емес,

BBh — қатты дискіні" белгісіз қатесі.

**INT 13h, 02h функциясы. Секторды оқу.**

Бір немесе бір топ секторларды физикалық дискіден (логикалық емес!) жадыға оқиды. Бастапқы сектор үшін абсолютті координаттар көрсетіледі (цилиндр, сектор, бастиек). Физикалық дискіні” секторлары әр жолда 1-ден бастап, цилиндрлер 0-ден бастап, бастиектер 0-ден бастап нөмірленеді. Алдымен 0 цилиндріні” 0 бастиегіні” 1...n-дейін секторлары, сосын 0 цилиндріні” 1 бастиегіні” 1...n дейін секторлары, содан со” 1 цилиндріні” 0 бастиегіні” 1...n дейін секторлары, т.б.с. ары қарай жалғаса береді. Осылайша 0 цилиндріні” 0 бастиегіні” 1 секторы бас жүктелу жазбасына (Master boot) жатады.

*Функцияны шақырғанда:* AH = 02h,

AL = оқылатын секторлар саны,

CH = цилиндр,

CL = бастапқы сектор (0...5 биттері) және цилиндр нөміріні”

екі үлкен биті (6-7 биті),

DH = бастиек.

DL = дискжетек:

00h...7Fh - иілгіш диск,

80h...FFh - қатты диск.

ES:BX = буфер адресі.

*Функциядан қайтарғанда:*

CF = 0,

AH = 0,

AL = оқылған секторлар саны.

*Қате кезінде:* CF = 1,

AH = күйі (00h функциясын қара).

**INT 13h, 03h функциясы. Секторды жазу.**

Бір немесе бір топ секторларды физикалық дискіге (логикалық емес!) жадыдан жазады. Бастапқы сектор үшін абсолютті координаттар көрсетіледі (цилиндр, сектор, бастиек). Физикалық дискіні” секторлары әр жолда 1-ден бастап, цилиндрлер 0-ден бастап, бастиектер 0-ден бастап нөмірленеді. Алдымен 0 цилиндріні” 0 бастиегіні” 1...n-дейін секторлары, сосын 0 цилиндріні” 1 бастиегіні” 1...n дейін секторлары, содан со” 1 цилиндріні” 0 бастиегіні” 1...n дейін секторлары, т.б.с. ары қарай жалғаса береді. Осылайша 0 цилиндріні” 0 бастиегіні” 1 секторы бас жүктелу жазбасына (Master boot) жатады.

*Функцияны шақырғанда:* AH = 03h,

AL = оқылатын секторлар саны,

CH = цилиндр,

CL = бастапқы сектор (0...5 биттері) және цилиндр нөміріні”  
екі үлкен биті (6-7 биті),

DH = бастиек.

DL = дискжетек:

00h...7Fh - иілгіш диск,

80h...FFh - қатты диск.

ES:BX = буфер адресі.

*Функциядан қайтарғанда:*

CF = 0,

AH = 0,

AL = жазылған секторлар саны.

*Қате кезінде:* CF = 1,

AH = күйі (00h функциясын қара).

**INT 13h, 05h функциясы. Иілгіш дискіде жолшықты пішімдеу.**

Көрсетілген жолшықты пішімдік та”баларды қойып  
пішімдейді.

*Функцияны шақырғанда:* AH = 05h,

AL = пішімделетін секторлар саны,

CH = цилиндр,

DH = бастиек,

DL = дискжетек (00h...7Fh),

ES:BX = адрестік өрістер тізімінің” адресі.

*Қате кезінде:* CF = 1,

AH = күйі (00h функциясын қара).

Ескерту.

Дискетті” жолшықты пішімдеу кезінде, адрес өрістеріні”  
тізімін құрастыру керек. Әрбір өріс 4 байттан тұрады. Оларда  
цилиндр, бастиек, сектор, сектор өлшемінің” кодасы көрсетіледі.  
Адрес өрістеріні” саны жолшықтағы секторлар санына те”.  
Сектор өлшемінің” кодасы келесі мәндерін қабылдай алады:

0 – секторда 128 байт

1 – секторда 256 байт

2 – секторда 512 байт

3 – секторда 1024 байт

PC/AT машиналарда бұл функцияны” алдында 13h үзуіні”  
17h функциясы орындалуы тиіс.

**INT 16h, 00h функциясы. Пернетақтадан символды оқу.**

Енгізу сақиналық буферден символды және скан-кодасын  
оқиды. Егер буфер бос болса, символды” енгізуін күтеді.

*Функцияны шақырғанда:* AH = 00h.

*Функциядан қайтарғанда:*

АН = скан-кодасы,  
AL = ASCII символ.

**INT 16h, 01h функциясы. Пернетақтаны” күйін алу.**

Сақиналық буферде күтіп тұрған символдар бар екенін анықтайды; күту жалауын және бар болса символ өзін қайтарады. Бірақ символ мен скан-кодасы сақиналық буферден алынбайды және 00h функциясын шақырғанда олар буферден оқылады.

*Функцияны шақырғанда:* АН = 01h.

*Функциядан қайтарғанда:* егер символ күтіп тұрса:

ZF = 0,

АН = скан-кодасы,

AL = ASCII символ.

Егер күтіп тұрған символдар болмаса:

ZF = 1.

**INT 16h, 02h функциясы. Пернетақтаны” жалауларын алу.**

Пернетақтаны” басқару пернелеріні” күйін сипаттайтын, пернетақтаны” жалаулар байтын қайтару (0000h:0417h адресі бойынша BIOS деректер аймағыны” байты).

*Функцияны шақырғанда:* АН = 02h.

*Функциядан қайтарғанда:* AL = жалаулар. Байтты” биттеріні” келесі мәндері бар:

- 0 – O” Shift пернесі басылып тұр,
- 1 – Сол Shift пернесі басылып тұр,
- 2 – Ctrl пернесі басылып тұр,
- 3 – Alt пернесі басылып тұр,
- 4 – Scroll Lock режимі қосылып тұр,
- 5 – Num Lock режимі қосылып тұр,
- 6 – Caps Lock режимі қосылып тұр,
- 7 – Insert режимі қосылып тұр.

## В ҚОСЫМШАСЫ ЖҮЙЕЛІК КЕСТЕЛЕР

В-1 кесте

**Байланысты” векторлық кестесіні” пішімі**

| <i><b>Ығысу</b></i> | <i><b>Байт саны</b></i> | <i><b>Сипаттама</b></i>                                                                                |
|---------------------|-------------------------|--------------------------------------------------------------------------------------------------------|
| -18h                | 2                       | AX=5E01h болғанда INT 21h шақыру кезінде CX-ті” мазмұны                                                |
| -16h                | 2                       | Бүркемелейтін FCB үшін санауыш                                                                         |
| -14h                | 2                       | Ашық FCB үшін санауыш                                                                                  |
| -12h                | 4                       | OEM-функциялар ө”деуішті” адресі (FFFFh:FFFFh, ө”деуіш орнатылмаған болса)                             |
| - 0Eh               | 2                       | DOS-ты” кодалық сегментінде INT 21h үзуінен қайтару кодасыны” ығысуы                                   |
| - 0Ch               | 2                       | Қайталауды” санауышы                                                                                   |
| - 0Ah               | 2                       | Қайталауды кідірту                                                                                     |
| - 08h               | 4                       | Ағымдағы дискіні” буферіне нұсқағыш                                                                    |
| - 04h               | 2                       | DOS-ты” деректер сегменті                                                                              |
| - 02h               | 2                       | Бірінші MCB-ді” (жады басқару блогыны”) сегменттік адресі                                              |
| 0h                  | 4                       | Диск параметрлер бірінші блогына (Device Control Block) нұсқағыш                                       |
| 4h                  | 4                       | Бірінші (SFT) жүйелік файлды” кестесіне нұсқағыш                                                       |
| 8h                  | 4                       | CLOCK\$ белсенді құрылғыны” тақырыбына нұсқағыш                                                        |
| 0Ch                 | 4                       | CON белсенді құрылғыны” тақырыбына нұсқағыш                                                            |
| 10h                 | 2                       | Бөлек блоктардағы мәліметтерді жіберуге арналған құрылғылар үшін блокты” максимальды көлемі (байттағы) |
| 12h                 | 4                       | Дискілік буферлерді сипаттайтын құрылымына нұсқағыш                                                    |
| 16 h                | 4                       | Құрылғылар туралы ақпарат массивіне нұсқағыш                                                           |
| 1Ah                 | 4                       | FCB кестесіне нұсқағыш                                                                                 |
| 1Eh                 | 4                       | FCB кестесіні” өлшемі                                                                                  |

|     |    |                                                                                                  |
|-----|----|--------------------------------------------------------------------------------------------------|
| 20h | 1  | Орнатылған блокты құрылғыларды” саны                                                             |
| 21h | 1  | Config.sys файлындағы LASTDRIVE мәні (келісім бойынша-5)                                         |
| 22h | 18 | NUL құрылғы драйверіні” тақырыбы. Бұл драйвер MS-DOS-тағы драйверлер тізімінде әрқашанда бірінші |
| 3Fh | 2  | BUFFERS=X,Y дерективадағы X мәні                                                                 |
| 41h | 2  | BUFFERS=X,Y дерективадағы Y мәні                                                                 |
| 43h | 1  | Ж <sub>3</sub> ктеу дисководы (1=A:)                                                             |
| 45h | 2  | Ke”ейтілген жадыны” килобайтпен өлшенген көлемі.                                                 |

В-2 кесте

### Ехе файл тақырыбыны” пішімі

| <i>Ығысу</i> | <i>Байт саны</i> | <i>Сипаттамасы</i>                                                                                         |
|--------------|------------------|------------------------------------------------------------------------------------------------------------|
| 00h          | 2                | Signature - Ехе файлды” белгісі (MZ символдарды” кодасы)                                                   |
| 02h          | 2                | PartPag - 512 модулі бойынша тақырыбы қосылған программаны” өлшемі                                         |
| 04h          | 2                | PageCnt — блогтағы 512 байт бойынша тақырыбы бар программаны” өлшемі                                       |
| 06h          | 2                | ReloCnt — адрестерді баптау (настройка) кестесіні” элементтер саны                                         |
| 08h          | 2                | HdrSize — параграф бойынша тақырыбыны” өлшемі (параграф=16 байт).                                          |
| 0Ah          | 2                | MinMem — жүктелген модульден кейін программаға бөліп беретін жадыны” минималды көлемі (параграф бойынша)   |
| 0Ch          | 2                | MaxMem - жүктелген модульден кейін, программаға бөліп беретін жадыны” максималды көлемі (параграф бойынша) |
| 0Eh          | 2                | ReloSS — стек сегментіні” программа басынан ығысуы (параграф бойынша)                                      |
| 10h          | 2                | ЕхеSP - программаға кіру кезіндегі SP регистрді” мазмұны                                                   |
| 12h          | 2                | ChkSum - бақылау қосындысы                                                                                 |
| 14           | 2                | ЕхеIP - программаға кіру кезіндегі IP регистрді” мазмұны                                                   |

|     |     |                                                                                             |
|-----|-----|---------------------------------------------------------------------------------------------|
| 16h | 2   | ReloCS — кодалық сегментіні” программа басынан ығысуы (параграф бойынша)                    |
| 18h | 2   | TablOff - адресстерді баптау (настройка) кестесіні” бірінші элементіні” салыстырмалы адресі |
| 1Ah | 2   | Overlay — оверлей нөмірі (түпкі модулі үшін - 0)                                            |
| 1Ch | ... | Айнымалы ұзындықты адресстерді баптау кестесі                                               |

3 кесте

### SFT-ны” тақырыбы

| <i>Ығысу</i> | <i>Байт саны</i> | <i>Сипаттамасы</i>                                                                  |
|--------------|------------------|-------------------------------------------------------------------------------------|
| 0            | 4                | Осы кесте со”ғы болғанда, бірінші сөзде FFFFh немесе келесі кестені” адресі болады. |
| 4            | 2                | Осы кестедегі файлдар сипаттау блогыны” саны.                                       |

В-4 кесте

### Файл сипаттау блогы

| <i>Ығысу</i> | <i>Байт саны</i> | <i>Сипаттамасы</i>                                                                                            |
|--------------|------------------|---------------------------------------------------------------------------------------------------------------|
| 00h          | 2                | Осы файлға бекітілген дескрипторлар саны немесе 0 (осы блок бос болса)                                        |
| 02h          | 2                | Файлды ашу кезіндегі қатынау (доступ) режімі                                                                  |
| 04h          | 1                | Файл атрибуттары                                                                                              |
| 05h          | 2                | Құрылғыны” ақпараттық сөзі                                                                                    |
| 07h          | 4                | Символдық құрылғыны” драйвер тақырыбына нұсқағыш немесе (файлдар үшін) дискжетекті” параметр блогына нұсқағыш |
| 0Bh          | 2                | Файлды” бірінші кластеріні” саны                                                                              |
| 0Dh          | 2                | Файлды” со”ғы өзгертуіні” уақыты                                                                              |
| 0Fh          | 2                | Файлды” со”ғы өзгертуіні” мезгілі (датасы)                                                                    |

|     |    |                                                                       |
|-----|----|-----------------------------------------------------------------------|
| 11h | 4  | Байт бойынша файлды” өлшемі                                           |
| 15h | 4  | Файл нұсқағышыны” ағымдағы орны                                       |
| 19h | 2  | Файлды” оқылған немесе жазылған со”ғы кластеріні” салыстырмалы нөмірі |
| 1Bh | 4  | Осы файл туралы каталогты” жазбасы бар сектор нөмірі                  |
| 1Fh | 1  | Секторды” ішіндегі каталог жазбаны” нөмірі                            |
| 20h | 11 | Файлды” аты және ке”ейтуі (жолсыз)                                    |
| 2Eh | 2  | Файлды ашқан программаны”, PSP-ды” сегменттік адресі                  |
| 35h | 2  | Файлды” оқылған немесе жазылған со”ғы кластеріні” абсолютті нөмірі    |

В-5 кесте

### PSP пішімі

| <i>Бйы-<br/>су</i> | <i>Байт<br/>саны</i> | <i>Сипаттамасы</i>                                                                          |
|--------------------|----------------------|---------------------------------------------------------------------------------------------|
| 0                  | 2                    | Int 20h командасы (программалар бұл команданы өзіні” жұмысын аяқтау үшін қолдана алады)     |
| 2                  | 2                    | Программаға бөліп берген жадыны” бірінші байтыны” сегменттік адресі                         |
| 4                  | 1                    | Қолданылмайтын байт толтырғыш                                                               |
| 5                  | 1                    | CP/M жүйелік шақыру                                                                         |
| 6                  | 2                    | COM-файл үшін бірінші сегмент                                                               |
| 8                  | 2                    | FAR JMP 05h сақталады                                                                       |
| 0Ah                | 4                    | INT 22h сақтайды (аяқталу адресі)                                                           |
| 0Eh                | 4                    | INT 23h сақтайды (Ctrl+Break ө”деушіні” адресі)                                             |
| 12h                | 4                    | INT 24h сақтайды (критикалық қатені ө”деуішті” адресі)                                      |
| 16h                | 2                    | Аталық PSP-ды” сегменттік адресі                                                            |
| 18h                | 20                   | JFT файлдарды” жұмыс кестесі. Файлға бір байттан (FFh болса, онда бұл кесте қолданылмайды). |
| 2Ch                | 2                    | Программа ортасы блогыны” сегменттік адресі                                                 |

|     |     |                                                                           |
|-----|-----|---------------------------------------------------------------------------|
| 2Eh | 4   | INT 21h үзуіні” со”ғы шақыруына кіру үшін программаны” SS:SP              |
| 32h | 2   | JFT файлдарды” жұмыс кестесіні” элементтер саны (келісім бойынша 20)      |
| 34h | 4   | JFT файлдарды” жұмыс кестесіні” адресі (келісім бойынша PSP:0018h)        |
| 38h | 4   | Алдағы PSP-ға нұсқағыш                                                    |
| 40h | 2   | 30h функция пішімі бойынша DOS-ты” версиясы                               |
| 42h | 26  | Қосалқыланған (MS Windows және 6.00 версиясынан жоғары DOS пайдаланылады) |
| 50h | 3   | INT 21h/RETf командалары (DOS-ты шақыру және қайта оралу)                 |
| 5Ch | 16  | Келісім бойынша бірінші FCB                                               |
| 6Ch | 16  | Келісім бойынша екінші FCB                                                |
| 7Ch | 4   | Пайдаланылмайды                                                           |
| 80h | 128 | Командалық жол/келісім бойынша DTA (зарезвированно).                      |

В-6 кесте

### Тегерішті” параметрлер блогы

| <i><b>Ығысу</b></i> | <i><b>Байт саны</b></i> | <i><b>Сипаттамасы</b></i>                                         |
|---------------------|-------------------------|-------------------------------------------------------------------|
| 00h                 | 1                       | N тегеріштер (00h=A; 01h=8; и т.д.).                              |
| 01h                 | 1                       | Драйверлі құрылғыны” ішіндегі N құрылғы.                          |
| 02h                 | 2                       | Секторды” байт бойынша мөлшері.                                   |
| 04h                 | 1                       | Кластердегі секторды” е” үлкен нөмірі (кластер мөлшері-1).        |
| 05h                 | 1                       | Кластерді секторға ауыстыруда қолданылатын жылжулар (сдвиг) саны. |
| 06h                 | 2                       | Дискіні” басында сақталған секторлар саны                         |
| 08h                 | 1                       | Файлдарды орналастыру кестесіні” саны.                            |
| 09h                 | 2                       | Түпкі (каталогтағы) тізбедегі элементтер саны.                    |
| 0Bh                 | 2                       | Деректер басталатын N секторлар.                                  |

## ӘДЕБИЕТТЕР ТІЗІМІ

1. Финогенов К.Г. Самоучитель по системным функциям MS DOS.- М.: МП "МАЛИП", 1993.
2. Рудаков П.И., Финогенов К.Г. Язык ассемблера: уроки программирования.- М.: ДИАЛОГ-МИФИ, 2001.
3. Пирогов В.Ю. ASSEMBLER. Учебный курс. – М.: "Нолидж", 2001.
4. Зубков С.В. Assembler для DOS, Windows и Unix. – М.: ДМК, 1999.
5. Трумов А.Ч., Куций А.П. ЭЕМ, кешендер, жүйелер және тораптарды” жүйелік программалық қамтамасы. - Алматы: ҚазҰТУ, 1995.
6. Тұрым А.Ш. Ақпараттану және есептеу техникасыны” түсіндірме сөздігі. –Алматы: ҚазҰТУ, 1998.
7. Қазақша – орысша, орысша – қазақша терминологиялық сөздік. 3 т. Информатика және есептеуіш техника. – Алматы: "Рауан", 1999.