

А 2011  
28792к

И.Т. Утепбергенов, А.И. Буранбаева

# ЖҮЙЕЛІК ПРОГРАММАЛАУ

(Оқу құралы)



Алматы  
2008

1 2011 / 28792 к

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

«М. Тынышпаев атындағы Қазақ көлік және коммуникациялар  
академиясы» АҚ

И.Т. Утепбергенов, А.И. Буранбаева

## **ЖҮЙЕЛІК ПРОГРАММАЛАУ**

**(Оқу құралы)**

Алматы

2008

004.45(075)

У 882

ЕОК 004.45 (075)

ББК 32.973 я 73

У82

Пікір жазғандар: Д.Н. Шукаев, т.ғ.д., профессор, ҚазҰТУ  
Р.Г. Бияшев, т.ғ.д., профессор, АжБМИ  
Б.С. Ахметов т.ғ.д., профессор, ҚазККА

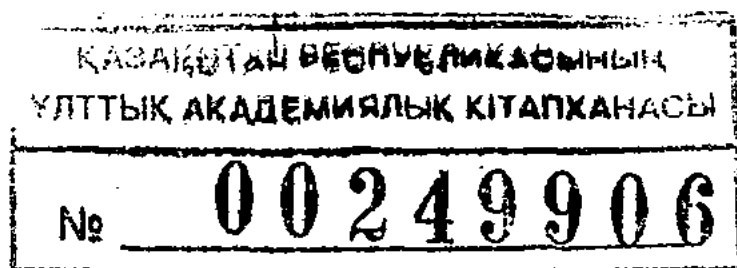
У82 Утепбергенов И.Т., Буранбаева А.И. Жүйелік программалау: Оқу құралы. - Алматы: ҚазККА, 2008, - 89 б.

ISBN 978-601-207-230-3

Оқу құралында техникалық мамандықтарының компьютерлік ақпараттық жүйелерді пайдалану артықшылықтарының ерекшеліктері жан-жақты қарастырылған. Сонымен қатар символдық программалау жүйелері сипатталып, Лисп программалау тілінің барлық командалары баяндалады.

ББК 32.973 я 73

*Баспаға авторлық нұсқада ұсынылды.*



ISBN 978-601-207-230-3

© «М. Тынышпаев атындағы ҚазККА» АҚ, 2008  
© Утепбергенов И.Т., Буранбаева А.И., 2008

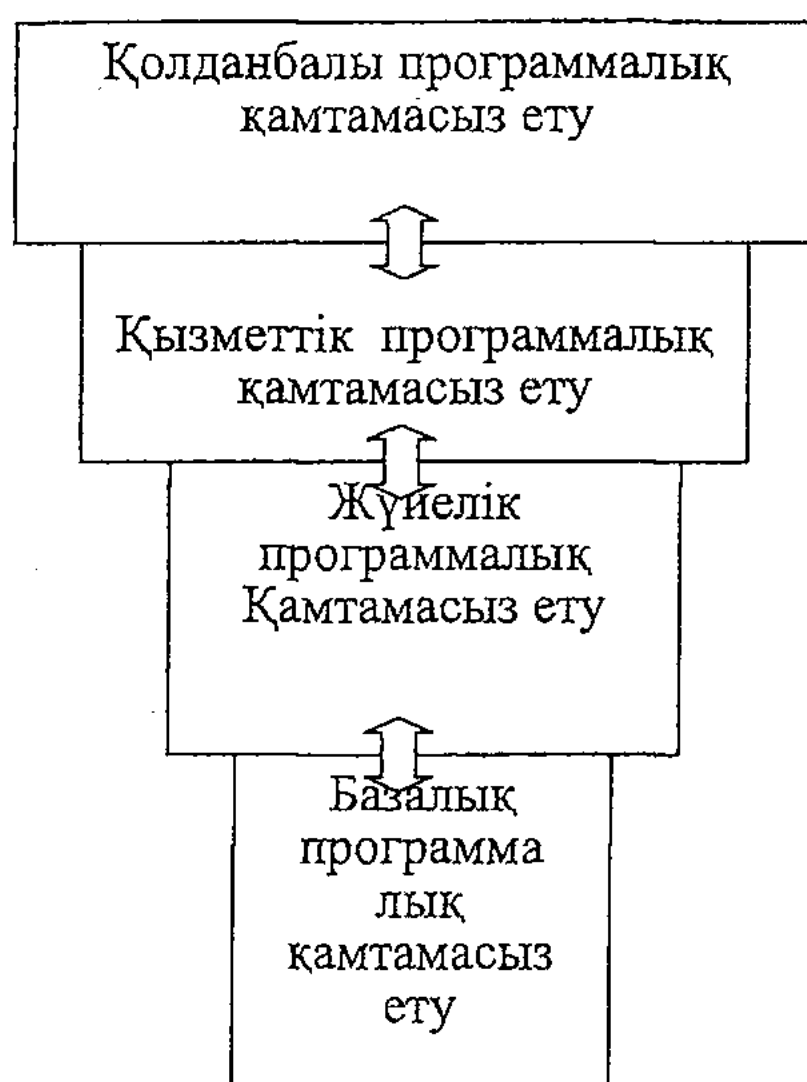
## КІРІСПЕ

Ассемблер тілінде программалау пәні Intel процессорлары негізіндегі компьютер архитектурасын оқыған кезде базалық пән болып табылады. Intel микропроцессорларының программалық моделдеріне түсініктемелер келтірілген. Сондай-ақ операндтарды адресациялау әдістері және ассемблер тіліндегі программа құрылымы түсіндірілген. Мұнда Ассемблер тілінің негізгі түсініктері, ассемблер тілінде программалаудың қазіргі заманға сай талаптары, соның ішінде DOS және Windows үшін жүйелік және қолданбалы программалау қарастырылған. Соның ішінде MS DOS ОЖ үшін жүйелік функциялар және Windows қосымшалар үшін API функциялардың түсініктері мен қолдану әдістері келтірілген. Пәннің өңделген оқу-әдістемелік кешені (СП ОӘК) студенттердің алдына «Ассемблер тілінде программалау» курсы сапалы және тиімді оқып шығу мақсатында жасалған. ОӘК дәрістік және зертханалық сабақтардан, бақылау сұрақтарының үйде орындауға берілген тапсырмалардан және СӨЖ тақырыптарынан тұрады. Бақылау сұрақтары мен үйге орындауға берілген тапсырмалар әрбір дәрістік сабақтан кейін келтірілген. Студенттерді қадағалап қорытынды бақылау жасау үшін, сондай-ақ ағымдық және аралық бақылаулардан сүрінбей өтуі мақсатында ОӘК сұрақтары келтірілген. Ұсынылып отырған әдебиет жаңа заманға сай өңделіп, студенттерге ұсынылды.

## 1. Программалық қамтамасыз ету

Бағдарламалық және аппараттық қамтамасыз ету компьютерде тығыз байланыста және үздіксіз қарым-қатынаста болады. Дегенмен де біз, бұл дәрежелерді екі түрлі қарастырамыз. Бұл екеуінің арасында диалектикалық байланыс бар екенін ұмытуға болмайды. Және де бұларды екі бөліп қарастыруымыз шартты.

Программалық қамтаманың сандық құралын – **программалық нұсқа** деп атайды. Программалар арасында, физикалық және блоктардың арасындағыдай қарым-қатынас болады. Сонымен қатар басқа да төменгі қатардағы бағдарламалармен жұмыс істейді. Ол дегеніміз, программалық интерфейс жайында. Бұндай интерфейстердің бар мүмкіндігі техникалық шарттардың және қарым-қатынастарының хаттамасы бар екендігіне негізделген. Ал тәжірибеде ол бағдарламалық қамтаманың бірнеше қарым-қатынастарға дәрежелерінің өзара бөлінуімен қамтамасыз етіледі.



Бағдарламалық (программалық) қамтамасыз ету дәрежелері өздігінен пирамидалық құрылымды көрсетеді.

### Базалық дәреже

Базалық программалық қамтама – ең төмен дәрежедегі программалық қамтама болып табылады. Ол базалық аппараттық құралдардың қарым-қатынасына жауап береді. Базалық программалық құрал өздігінен базалық құрал-жабдықтардың құрамына кіреді. Және арнайы микросызбаларда сақталады. Олар тұрақты естесактау құрылғысы деп аталады (ТЕҚ – ПЗУ – Read Only Memory, ROM). Программалар және берілген ТЕҚ-ның

микросызбасында жазылады. Өндіріс кезеңінде бұларды пайдалана отырып өзгертулер енгізуге болмайды.

Базалық программалық құралдарды пайдалану кезінде өзгеруі техникалық жағынан орынды. ТЕҚ (Тұрақты естесактау құрылғысы) микросызба орнына жаңа программалық әрқашан естесактау құрылғысын (ЖПТЕҚ – Erasable and Programmable Read Only Memory, EPROM) қолданылады. Бұл жағдайда ТЕҚ мазмұнының өзгеруін сандық жүйенің құрамында орындауға болады. (Бұл өндірісте – Флеш-өндіріс деп аталады). Одан басқа жүйелерде де арнайы құрылғыларда да, программаларда да солай аталады.

### **Жүйелік дәреже**

Жүйелік дәреже - өтпелі кезең болып табылады. Бұл жүйеде жұмыс атқаратын программалар компьютер жүйесіндегі программалардан басқа, яғни базалық дәрежедегі программалармен байланысып аппараттық қамтамасыз ету арқылы анықтайды. Бұл дегеніміз ол “делдалдық” қызмет функциясын орындайды дегенді білдіреді.

Бұл дәрежедегі программалық қамтама, сандық жүйенің пайдаланушылық көрсеткішінің белгісі болады. Осылай, мысалы, сандық жүйеге жаңа құралдар көмегімен қосылу кезінде жүйелік дәрежеде программа құруға болады. Ол программа осы құралмен басқа да программалардың байланысып отыуын қамтамасыз етеді. Нақты программалар нақты құрылғылармен қарым-қатынаста жауап береді. Құрылғы драйвер деп аталады. Олар жүйелік дәрежедегі программалық қамтаманың құрамына кіреді. Басқа дәрежелік жүйенің программалық кластары (топтары) оның қолданылуы мен қарым-қатынасын анықтайды. Осылардың көмегімен ол берілген мәліметтерді санақтық жүйеге енгізу мүмкіндігіне ие болады. Сонымен қоса оның жұмысын басқару және өзіне қолайлы формада нәтиже алады. Бұл программаның құралдары қолданушылар интерфейсінің қамтамасыз ету болып табылады. Компьютердегі жұмыс қолайлылығы және жұмыс орнындағы еңбек өнімділігі осы программаларға байланысты болады.

Қызметтік программа құралдарының сұрыпталуы. Оларға жататындар:

- Файл диспетчерлері;
- Мәлімет жинау амалдары;
- Көру және есту амалдары;
- Диагностикалық амал;
- Бақылау амалы;
- Орнату мониторлары;
- Қатынас амалдары;
- Компьютер қауіпсіздігін қамтамасыз ету құралдары.

**Файл диспетчерлері** (файлдық менеджерлер). Мұндай кластағы программалар көмегімен көптеген операциялар, файл құрылыстарымен қызмет жасауға байланысты файлдарды: көшіру, орналастыру және атын

өзгерту, каталог құру (папка) файлдарды және каталогтарды өшіру, файл құрылысының ашылуы және файлдарды іздеу сияқтылар жатады. Базалық программалық амалдар, бұл бағытқа арналған, әдеттегі жүйелік дәрежедегі программалар құрамына кіреді. Және операциялық жүйемен бірге орнатылады. Компьютермен жұмыс істеу қолайлылығын жоғарылату үшін көптеген тұтынушылар қосымша қызметтік программалар орнатып алады.

#### **Мәлімет жинау амалдары (архиваторлар).**

Архив құруға арналған. Архивтелген мәліметтер, олардың сақталуын, бір архивтік файлға көптеген каталогтар және файлдар топтарын енгізу арқылы байланыстыру. Архиваторларды бағалы мәліметтер қорын құру негізінде көптен қолданады.

**Көру және есту амалы.** Әдетте файлдармен жұмыс істеу кезінде, “тудырушы” қолданбалы жүйесіне енгізу керек. Олар осының көмегімен пайда болады. Ол құжаттарды көруге өзгерістер енгізуге мүмкіндік береді.

**Диагностикалық амал.** Диагностикалық процестерді автоматтандыруға және аппараттық қамтамасыз етуге арналған. Олар керекті тексерулер жүргізіп және жиналған ақпаратты қолайлы түрде береді. Оларды тек бұзылудан сақтау үшін емес, сонымен қоса компьютер жұмысын жеңілдету үшін қолданады.

#### **Басқару амалдары (мониторингтік).**

Программалық басқару амалдарын кейде – мониторлар деп атайды. Олар компьютер жүйесіндегі процесті бақылауға мүмкіндік береді. Бұл жағдайдың екі түрлі жолы қарастырылған: шынайы тәртіпте уақытпен бақылау немесе арнайы файлдық хаттамадағы жазбалар қорытындысына бақылау жасау.

Бірінші жолы – санау жүйесінің жұмысын жеңілдету жолдарын іздеу мақсатында және оның жұмысын жақсарту мақсатында қолданады.

Екінші жолында тек мониторинг автоматты түрде қашықтан басқарылған кезде қолданамыз. Соңғы жағдайда мониторингтің қорытындысын техникалық қолдау қызметтігіне беруге болады.

Мониторинг амалдары, шынайы уақыт тәртібінде жұмыс істейді. Әсіресе компьютер жұмыс істеген кезде тәжірибелік жолмен істелінеді. Өйткені кейбір процесстер қолданушылардың көзіне көрінбеуі мүмкін. Бұл жағдай сол процесстерге арналған.

**Орнату мониторлары.** Бұл категорияның программалары тек программалық қамтамаларды бақылау үшін қолданылады. Берілген программалық қамтаманың керектігі ол дер кезінде әр түрлі категориядағы программалар арасында байланыс орнайды.

Орнату мониторлары программалардың өзгеру жағдайын, оған әсер етушілерді бақылайды. Сонымен қатар тіптен бұрын орнатылған кейіннен жойылып кеткен программаларды қайта қарауға мүмкіндік береді.

Қондырғыны басқарудың ең оңай амалы және программаны жою, әдетте операциялық жүйенің құрамына кіреді және программалар қамтамасының жүйелік дәрежесінде орналасады. Бірақ олар көп жағдайда жеткіліксіз

болады. Сондықтан санау жүйесін олар жоғары дәрежелі қауіпсіздікті қажет етеді, қосымша қызметтік программаны пайдаланады.

**Қатынас амалдары** (коммуникационные программы). Электрондық байланыстың пайда болуымен және компьютер желілерінің пайда болуына байланысты бұл бағдарламалардың маңыздылығы арта бастады. Олар алыс қашықтықтағы компьютерлермен байланыс құруға мүмкіншілік тудырады. Электрондық пошталардың хат алмасуына қызмет етеді. Телеконференциялық жұмыстарды атқарады. Сол сияқты басқа да компьютер желісіндегі операцияларға қатысады.

**Компьютер қауіпсіздігін қамтамасыз ету амалдары.**

Бұл категорияға көп таралған активті және пассивті амалдар жатады. Олар компьютерді бұзылудан сақтайды.

Пассивті қорғаудың амалы ретінде қызметтік программалар қолданылады. Олар көп жағдайда базалық қызметтерге ие.

Активті қорғаудың амалы ретінде антивирустік программалар қамтамасыз жатады.

Информациялық және математикалық қамтама санау жүйесі жайында түсінік.

Аппараттық және программалық қамтамамен санау техникасының амалдарын кей жағдайларда информациялық қамтама ретінде қарастыру керек. Бұл берілген программалар жұмысына қажет болып табылады.

Арнайы компьютер жүйелерінде (ракеталар, самолеттер, космостық ұшу аппараттың және т.б.) программалық аппараттық қамтаманы математикалық қамтама деп атайды. Әдетте, ол “қатты” микросызбаларда жазылады; яғни ПЗУ және тек ПЗУ-да өзгерту арқылы өзгертіледі немесе арнайы жабдықпен қайта программалау арқылы өзгерте аламыз.

### **UNIX операциялық жүйесінің түрлері**

UNIX операциялық жүйесінің мүмкіншіліктері және әртүрлілігі, LINUX-тің әртүрлілігі.

**Операциялық жүйе** – бұл қабылдаулар және процедуралар комплексі аппараттық амалдардың бағдарлама машиналарын жіберу және басқару жұмыстарын қамтамасыз етеді. Басқа сөзбен, персоналды компьютердегі жұмыс тәртібінің ортақ комплексі. Операциялық жүйенің бірнеше типі белгілі. Соның ішінде көп тараған жүйе Microsoft компаниясы. Оларды қолдану үлкен жүйелік талаптарды, мысалы оперативті есте сақтаудың көлемін айтамыз. Мысал ретінде 1988 жылы оперативтік есте сақтау талаптардың өзгеруін қарастырайық.

Операциялық жүйе жүйелік және қызметтік программалардың комплексін білдіреді. Бір жағынан ол компьютердің базалық программалық қамтамаға сүйенеді, ол жүйенің құрамына кіретін BIOS, басқа жағынан ол өте жоғары дәрежелі программалық қамтамаларға басқарушы болады. Ол тек қолданбалы және көптеген қызметтік программалар.

Операциялық жүйенің енгізілуін мына программаларды, берілген жүйенің басқару жұмысына арналған программаларды жатқызамыз.



Барлық операциялық жүйенің негізгі функциясы – делдалдық. Ол бірнеше түрлі интерфейспен қамтамасыз етумен бекітіледі.

Тұтынушы интерфейсімен және компьютер программасының аппараттық амалдары, программалық интерфейс және апараттық амалдармен жатқызамыз, әртүрлі интерфейс программаларын қамтамалау түрлері.

UnixWare SVR 4-ке негізделген. Бірақ уақыт жүйесі Novell бақылаған, ал қазір SCO UNIX WARE жұмыстарды жүргізеді. x86 архитектурасында тек машиналарда жұмыс істейді.

BSDI internet server – коммерциялық дистрибутив BSDIOS. Ол бастан-ақ Беркли университетінде жасалып шығарылған болатын. BSDI өз негізі етіп BSDIOS және жаңа утилиттерді қосты, бағдарлады және функцияларда бұл жүйенің және даму түсінуіне әсер етті Internet – шлюзде.

Free BSD және Net BSD – ақысыз альтернативтер BSDI Internet server. Бұл жүйелер көптеген функцияға BSDIOS-тың танымалдығын қамтамасыз ететін, бірақ оларда мамандандырылған техникалық қолдау жетіспейді, кез-келген коммерциялық шығарылымға тән.

SCO Open server – тағы да бір түрі UNIX-тің Intel үшін. Ол корпоративті желілерде өте танымал және көп жылдар бойы жұмыс істеп келеді. Жүйе жоғарғы орындарға ие, корпоратив желілерде өте танымал және көп жылдар бойы жұмыс істеп келеді. Жүйе жоғарғы орындарға ие, корпоратив серверлері арасында, дегенмен оған BSD әмбебеп жетіспейді.

Linux – ақысыз операциялық жүйе идеясының дамуы кезінде Minix-те қолданылған. Бұл өкім кейіннен жиі қолданылатын, танымал және қолдауға ие операциялық жүйе болады. Linux, Posix және SSV стандарттарына негізделген Linux тұтынушылар арасында Unix-те жұмыс істейтін өте танымал ол жеңіл орнатылады.

Linux. Операциялық жүйелер базарында Microsoft (OS Mac, OS/2, Linux) компаниясының бақталасы бола алады.

Жаңа Linux-тің қасиеттері келесі түрде: біріншіден, ол ақысыз таратылады; екіншіден, Linux қатты дискіде аз орын алады; үшіншіден, Linux үзіліссіз жылдар бойында жұмыс істей алады. Linux-тің таралуына ең үлкен кедергі – қолайлылықтың жоқтығы. Linux-ті қосу және орнату Windows-қа қарағанда қиын. Егер өндірушілер Windows персоналды компьютерге ешқашан перифериясының қосымша қажеттіліктерін шығармаса, онда Linux-ке арналған драйверлер үшін программистер тәуелсіз программалар жаза берер еді. Linux-ке көп жүйелер қажет. Сондықтан бұл көп уақытты қажет етеді.

Мысалы, Linux, USB перифериясын әлі күнге шейін қолдамайды. Бірақ уақыт өте келе жағдай өзгереді. Енді кейбір өндірушілер Linux-ке өз продукциясын ұсынып отыр.

## 1.2 Деректердің берілуі

Бастапқы екі дәрістік сабақтардың мақсаты – Бірінші программа жазу үшін, ассемблер тілінің керекті және негізгі элементтерін беру. Соның ішінде

айнымалылар мен тұрақтылар, арифметика, есептеу тізбегін басқару, қарапайым енгізу – шығару функциялары.

Компьютердегі ақпараттың ең аз өлшем бірлігі - **бит** болып табылады. Бит 0 немесе 1 мәнін қабылдайды.

**8 биттен** тұратын топ **байтты** құрайды. Биттер немесе байттағы разрядтар оңнан солға қарай 0-ден 7-ге дейін нөмірленеді:

Биттер нөмірі:								
Биттер мәндері:								

Байт таңбасыз 0-ден 255-ке дейінгі және таңбалы - 128-ден +127-ге дейінгі сандардан тұруы мүмкін.

Екі байт немесе он алты байт сөз құрайды. Сөздегі биттер оңнан солға қарай 0-ден 15-ке дейін нөмірленеді:

Сөз таңбасыз 0-ден 65535-ке дейінгі және таңбалы -32768-ден +32767-ге дейінгі сандардан тұруы мүмкін.

Төрт байт немесе екі сөз екілік сөз құрайды, ол 32 биттен тұрады, ал екі екілік сөз төрт сөз құрайды (64 бит).

Компьютердегі бүкіл есептеулер екілік санау жүйесінде жүргізіледі, яғни екілік код арқылы. Сандарды онақтылық түрде көрсету, сандардың екілік кодтағы қысқа түрдегі жазуы болады. Ол үшін төрт екілік разрядты бір онақтылық разрядпен жазуға болады. Сандарды көрсетудің бұл екілік форматынан басқа ондық, екілік-ондық (BCD), және ASCII - код түрлері пайдаланылады.

#### **Екілік код.**

Екілік кодта екілік екі цифр бар. Ол 0 және 1. Екілік санының белгісі ретінде B әрпі қолданылады. Ол сан соңына қойылады. Мысалы: 10101001B.

Екілік санның мәні әрбір биттің орналасқан позициясы мен ондағы бірлік биттердің бар не жоғы арқылы анықталады.

Мысалы:

$$10001010b = 1*2^7 + 0*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0 = \\ = 128 + 8 + 2 = 138.$$

#### **Ондық код.**

Ондық кодта 0-ден 9-ға дейінгі сандар ондық цифрлар бар. Ондық санның белгісі болып D әрпі көрсетіледі. Ол санның ең аяғына қойылады. Егер санның соңында әріп болмаса онда сан ондық деп қабылданады.

#### **Онақтылық код.**

Ондық кодта 0-ден 9-ға дейінгі сандар және A-дан F-ке дейінгі әріптер болады. Онақтылық санның белгісі болып H әрпі көрсетіледі. Ол санның ең аяғына қойылады. Сан міндетті түрде цифрдан басталуы керек. Байт құрамында екі онақтылық разряд, сөз құрамында төрт разряд, екілік сөз құрамында сегіз разряд бар. Мысалы:

$$1BA8h = 1*16^3 + 11*16^2 + 10*16^1 + 8*16^0 = 4096+2816+160+8 = 708$$

Төменде 0-ден 15-ке дейінгі екілік, ондық, онақтылық сандардың мәндері берілген:

Екілік	Ондық	Он алтылық	Екілік	Ондық	Он алтылық
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	10	A
0011	3	3	1011	11	B
0100	4	4	1100	12	C
0101	5	5	1101	13	D
0110	6	6	1110	14	E
0111	7	7	1111	15	F

### Екілік-ондық код (BCD).

Екілік-ондық сандар оналтылық цифрлармен жазылады. Екілік-ондық сандар жинақталған немесе жинақталмаған форматта көрсетілуі мүмкін. Жинақталған форматта байт құрамында екі ондық цифр болуы мүмкін: 00-ден 99h дейін.

Жинақталмаған форматта байт құрамында кіші тетрадада бір цифр, ал үлкен тетрада 0-ге тең болады: 00-ден 09h дейін.

BCD кодтары екілік – ондық арифметика командаларында қолданылады  
**ASCII - код.**

Микропроцессор мен принтер, пернетақта немесе дисплей арасындағы деректермен өзара алмасу үшін ASCII - коды пайдаланылады. ASCII - коды (информация алмасуға арналған американдық стандартты код) компьютердегі алфавитті-цифрлық информацияны кодтауға арналған.

Мысалы 0-ден 9-ға дейінгі цифрларда 48-ден 57-ге дейін (30h тан 39h) ASCII – кодтары бар. “A” дан “Z” дейінгі бас әріптер 65-тен 90-ға дейін (41h тан 5Ah) ASCII - кодтары, ал “a” дан “z” кіші әріптері 097 ден 122-ге дейінгі (61h тан 7Ah) ASCII - кодтары бар. Алған нәтижені экранға шығару үшін оны ASCII - кодында көрсету керек. Мысалы: экранға 17h санын шығару керек дейік. Ол үшін алдымен оны ашу керек, сосын ASCII - кодына аудару керек:

17h → 0107h → 3137h

**Теріс сандарды көрсету.**

Теріс сандар компьютерде қосымша код арқылы көрсетіледі.

Санның таңбасын өзгерту үшін терістеуді орындайды, яғни екілік жүйеде берілген барлық сандардағы нөлдерді бірмен және бірлерді нөлмен ауыстырады да, одан кейін оған бір қосады.

Мысалы:

100 = 64h =            0110 0100b            санның тікелей коды  
терістеу            →    1001 1011b            санның кері ауысқан коды  
+1                    →    1001 1100b = 9Ch            санның қосымша коды

Қосымша кодты алу операциясын екіге дейін толықтыру операциясы деп атайды. Бұл форматтағы үлкен (7-ші, 15-ші, 31-ші байт , сөз, екілік сөз

үшін) биттер әруақытта санның таңбасын көрсетеді: 0 – оң сандар және 1 – теріс сандар үшін.

### **Microsoft фирмасының ассемблер тілінде программаны дайындау процесі**

Программаны дайындау және жөндеу процесі келесі этаптардан тұрады:

– Программа тексті кез – келген тексттік редактор арқылы дайындалады. Орындалатын тексттегі файл кеңейтілуі .ASM түрінде болуы міндетті;

– Программа трансляциясы MASM.EXE Ассемблерінің көмегімен, объектілік файл алу үшін орындалады;

– Объектілік модулдің компоновкасы LINK. EXE компоновщикі арқылы, жүктемеленетін (орындалатын) файл алу үшін орындалады;

– Дайын программаның орындалуы CodeView (CV.EXE файлы) интерактивті жөндегіш арқылы орындалады.

Программаның орындалатын текстін жазатын редактор таңдау кезінде ескеретін жағдай, көптеген тексттік процессорлар шығу файлына көмекші ақпарат қосып жібереді (мысалы, Microsoft Word). Сондықтан редактор таңдау кезінде ешқандай басқару символдарынсыз «таза текстін шығаратын редактор таңдаған жөн. Бұл редакторларға бізде кеңінен тараған Лексикон, Norton Editor сияқтылар жатады. Егер орындалатын тексттегі файл программасының аты P/ASM деп аталған болса, онда ассемблерді шақыру жолы келесідегідей түрде болады:

MASM /Z /ZI /N P, P, P;

/Z кілті программаның орындалатын текстіндегі, ассемблер тапқан қателерді экранға шығаруға рұқсат береді (бұл кілтсіз қателерді табуды трансляция листингі бойынша орындауға тура келер еді).

/ZI кілті объектілік файлға программаның орындалуы кезінде сұралмайтын, CodeView жөндегішімен қолданылмайтын орындалатын программа жолдарын және басқа ақпараттарды қосуды орындайды.

/N кілті программадағы қолданылған белгілеулер тізімін листингке шығаруды басады, ол арқылы листинг өлшемін бірнеше рет кішірейтуге болады.

Одан кейін орналасқан параметрлер модул аттарын береді: (P.ASM) орындалатын, (P. OBJ) объектілік және (P.LST) листинг файлдары. Ал нүктелі үтір қиылысқан сілтемелерді, яғни P.CRF файлын қалыптастыруды болдырмау үшін пайдаланылады.

Компоновщикті шақыру жолы келесідегідей түрде болады:

LINK /CO P, P;

/CO кілті орындалатын файлдағы CV жөндеуішінде орындалатын программаның толық текстін экранға шығару үшін символдық ақпараттарды береді, оған таңбалар, коментарилер және т.б. жатады. Одан кейін орналасқан параметрлер модул аттарын береді: (P. OBJ) объектілік және (P.EXE) жүктемеленетін файлдар. Ал нүктелі үтір (P.MAP) листингі

компоновкалау файлының қалыптастыруын болдырмау үшін және ішкі программа модулдерімен қолданылатын кітапханалық файлдарды пайдаланбау үшін қоданылады.

Компоновщик .EXE форматындағы жүктемелегіш файл құрады.

Егер орындалатын файл .COM форматында жазылған болса, онда трансляция және компоновкадан кейін қарапайым түрде оны .COM типіндегі файлға ауыстыру керек. Бұл үшін DOS құрамына кіретін EXE2BIN сыртқы командасы қолданылады:

EXE2BIN P P.COM

Бірінші параметр EXE2BIN командасына берілетін P.EXE жүктемеленетін файл екендігін көрсетеді, ал екіншісі ауыстырылған кездегі нәтиже. Екінші параметрдегі .COM кеңейтілуін көрсету міндетті түрде болу керек, өйткені үнсіздікпен EXE2BIN командасы .BIN кеңейтуіндегі файл құрады.

#### **Бақылау сұрақтары:**

1. Деректерді, сандарды беру кезінде қандай код түрлері қолданылады?
2. Кез-келген жады ұяшығының 20-разрядтық адресі қалай есептелінеді?
3. Ассемблер сөйлемдерінің түрлері?
4. Ассемблер тіліндегі программаны дайындау этаптары?

## 2. IBM PC архитектурасының ерекшеліктері

INTEL фирмасының микропроцессорлар тобына қысқаша шолу.

Кез-келген микропроцессордың маңызды характеристикасы ретінде деректер мен сыртқы шина адрестерін және оның ішкі регистрлерінің разрядтылығын айтуға болады. 8086 МП 16-разрядты ішкі регистрлер архитектурасынан және 16-разрядты деректер шинасынан тұрады. Сонымен, микропроцессор жұмыс жасай алатын (дерек немесе адрес), максимальды бүтін сан,  $2^{16}-1=65535$  (64К-1) құрайды. Бірақ 8086 МП адрестік шинасы 20 сызықтан тұрады, ол адрестік кеңістіктің  $2^{20}=1$  Мбайтқа сәйкес. 16-разрядтық адрестің көмегімен 20-разрядты адрестік кеңістіктің кез-келген нүктесіне қатынау үшін, микропроцессорда жадыны сегменттік адрестеу әдісі қарастырылған, олар төрт сегменттік регистрлермен таратылады.

Сегменттік адрестеудің негізі келесіде. Жадыда орындалатын кез-келген 20-разрядты адрес процессормен, осы ұяшық орналасқан, жады сегментінің бастапқы адресі мен сегменттің басынан бастап есептелген ығысуын, әдетте салыстырмалы адрес деп аталады, қосу арқылы анықталады. Төрт кіші битсіз яғни 16 бөлінген сегменттік адрес, сегменттік регистрлердің бірінде сақталады. Орындалатын адресі есептегенде процессор сегменттік регистрдің құрамын 16 көбейтеді (солға 4 екілік разрядқа жылжыту арқылы) және алынған 20-разрядтық адреске салыстырмалы адресі қосу арқылы алады. Базалық адресі 16 көбейту адрестік ұяшықтар көлемін ұлғайтады, яғни  $64 \text{ Кбайт} * 16 = 1 \text{ Мбайт}$  алуға болады.

80286 МП, IBM PC/AT компьютерларында орталық процессор ретінде қолданылғандықтан, 8086 МП қосымша жадыны басқару және оны қорғау схемалары бар күшейтілген варианты болып табылады. 80286 МП 16-разрядтық операндалармен жұмыс жасайды, бірақ 24-разрядтық адрестік шинасы бар, сәйкесінше  $2^{24}=16$  Мбайт адрестік кеңістікті пайдалана алады. Бірақ айта кететін жай бұл жадыны сегменттік адресациялау әдісі кезінде 1 Мбайт өлшемнен шығуға болмайды. Бұл шектеуді жеңу үшін 80286 МП (сондай-ақ 80386 МП) екі режимді қолдануға болады: нақты адрес режимі және қорғалған виртуальды адрес режимі немесе жай қорғалған адрес режимі. Нақты режимде 80286 МП алдында айтылғандай, 8086 МП сияқты функцияланады, онда да жылдамдығы жоғары және тек 1 Мбайт адрестік кеңістікке ғана қатынаса алады. Ал қалған 15 Мбайт жады аймағы компьютерде орнатылған болса да қолданысқа берілмейді.

Қорғалған режимде айтылғандай сегменттер және олардағы ығысулар қолданылады, бірақ бастапқы адрес сегменттік регистрлерді 16 көбейту арқылы емес, сол сегменттік регистрлермен индекстелетін, сегменттік регистрлердің дескрипторларының кестесінен. Әрбір сегменттік регистр 6 байт орын алады, олардың ішінде 3 байт (24 екілік разряд) ішкісегмент адресіне жатады. Ол арқылы толық 24-разрядтық адрестік кеңістікті қолдануға болады.

Әрбір сегменттік регистрдегі сегменттік дескрипторлар кестесінің индексіне 14 екілік разряд бөлінеде. Адрестелетін ұяшықтың толық логикалық адресі 14-разрядтық сегменттің индекс нөмірінен және 16 - разрядтық салыстырмалы адрестен тұрады. Бұл әрбір программа  $2^{30}=1$  Гбайт логикалық, немесе виртуалдық аймақты қолдана алу үшін арналған, сонымен, мүмкін болатын физикалық жады көлемін 64 рет көбейте аламыз. Виртуалды жадының операциялық жүйесі барлық орындалатын программалардың сегменттерін үлкен дискілік аймақта сақтайды, сегменттер керек болған жағдайда автоматты түрде оларды жедел жадыға жүктемелейді.

80386 және 80486 МП 32-разрядтық ішкі құрылымы және 32-разрядтық деректер шинасы және адрестері бар жоғарғы өнімді процессорлар болып табылады. Бұл деректерге сүйенсек, процессордың ішінде орналасқан регистрлердің, алдыңғы ертеректегі моделдердегі регистрлерден айырмашылығы, биттерінің ұзындығы 31-бит. Сондықтан микропроцессор жұмыс істей алатын максимальды бүтін сан  $2^{32}-1=4294967296$  (4Г) құрайды. Көп жағдайда 32-биттік операндтарды қолдану есептеуді жеделдетеді және жеңілдетеді. Сонымен жол, 80386 және 80486 МП регистрлер құрамы ұлғайтылған, ол да программалаушының жұмысын ыңғайлы ету үшін жасалған. Қорыта келе, процессорлардың жаңа модельдерінде көпесептік режимді, сондай-ақ көппроцессорлық жүйені қолдайтын құрылғылар орнатылған. Бұл микропроцессорлар да, 80286 МП сияқты, нақты және қорғалған режимде жұмыс жасайды. Біз қараған соңғы модельдегі микропроцессорлар  $2^{32}=4$  Г байт физикалық жадыны және  $2^{46}=64$  Г байт виртуалды жадыны адрестей алады. Мұнымен қоса айта кететін жағдай, өңдеушілерге ескі модел процессорлары мен жаңа модел процессорлары арасындағы толық келісу жағдайлары да жасалған, яғни, 8086-80286 процессорлары үшін жазылған 16-биттік операндтарды қолданған программаларды ешқандай жөндеусіз жаңа моделдегі процессорларда орындауға болады.

#### Адрестік кеңістіктің таратылуы

Дербес компьютер модификациясына және оның құрамындағы сыртқы құрылғыларға байланысты адрестік кеңістіктің таратылуы өзгешеленуі мүмкін. Бірақ жүйенің негізгі компоненттерінің орналасуы катал қадағаланады. Компьютердің адрестік кеңістікті қолданудың қарапайым кестесі 1-суретте келтірілген. Бұл суреттегі адрес мәндері 16-лық есептеу жүйесінде берілген. 16-лық сан екендігін, саннан кейін орналасқан h әріпіне байланысты білуге болады.

1 Кбайт	Үзу векторлары	00000h	
256 байт	BIOS деректер аймағы	00400h	
512 байт	DOS деректер аймағы	00500h	
	IO.SYS және MS		

	DOS.SYS		
	Жүктемеленетін драйверлер		> Стандартты жады (640Кбайт)
	COMAND.COM (резиденттік бөлім)		
	Жүктемеленетін қолданбалы және жүйелік программалар үшін арналған бос жады		
64 Кбайт	Графикалық буфер EGIA	A0000h =	
32 Кбайт	UMB	B0000h	
32 Кбайт	Текстік буфер EGIA	B8000h	
64 Кбайт	ПЗУ – BIOS ұлғайтулары	C0000h	>Жоғарғы жады (384 Кбайт)
64 Кбайт	UMB	D0000h	
128 Кбайт	ПЗУ BIOS	E0000h	
64 Кбайт	HMA	100000h =	
DOS 15 Мбайт (80286) 4 Гбайт-қа дейін (80386/486)	XMS	10FFF0h	>Ұлғайтылған жады

1-сурет. Адрестік кеңістікті тарату кестесі

Бірінші 640 Кбайт 00000h-тан 9FFFFh-қа дейінгі адресстердегі, адресстік кеңістікті негізгі жедел жады аймағына қалдырады, мұны тағы да стандартты жады аймағы деп те атайды (conventional). Жедел жадының бастапқы килобайты ұзу векторларымен толтырылған (4байттан 256 вектор бар). Ұзу векторларының артынан BIOS деректер аймағы орналасады, олар 00400h - тан 004FFh-қа дейінгі адресстерде орналасады. Бұл аймақта сыртқы құрылғыларды басқару процесінде BIOS программаларымен қолданылатын әртүрлі деректер орналасады, сонымен мұнда орналасатындар:

- жүйе көрсеткішімен кіріс клавиатура буфері;
- параллельді және тізбекті порттар адресі;



- видео жүйені құруға керекті деректер (курсор формасы және оның экрандағы сол кезде орналасқан орны, ағымдағы видеорежим, экран ені және т.б.);

- ағамдағы уақытты санау ұяшығы;

- есеп аралық байланыстар аймағы т.б.с.с.

BIOS деректер аймағы компьютерді бастапқы жүктемелеу процесі кезінде ақпараттармен толтырылады және керектігіне қарай динамикалық түрде жүйемен модификацияланады; көптеген қолданбалы программалар бұл аймаққа онда сақталған ақпараттарды оқу немесе модификациялау үшін қатынайды.

DOS жайындағы жүйелік деректер, жады аймағының 500h адресінен бастап орналасады. Сәйкесінше DOS деректер аймағынан кейін IO. SYS және MSDOS.SYS (IBMBIO.COM және PC-DOS жүйесі үшін IBMDOS.COM) файлдарымен жүктемеленетін операциялық жүйе орналасады және жүйе ондаған Кбайт орын алады.

Жоғарыда айтылып кеткен операциялық жүйе компоненттері әдетте 60-90 Кбайт орын алады. Одан кейінгі 640 Кбайт шекарасына дейінгі жады (кейде транзиттік аймақ деп те аталады) қолданбалы программаларды немесе кез-келген жүйені жүктемелеу үшін бос болады. Ал қалған 384 Кбайт адрестік кеңістік, жоғарғы (upper) жады деп аталады, тұрақты сақтау құрылғыларын (ТСК) орналастыру үшін арналған. Негізінен ТСК ретінде адрестердің кей бөліктері ғана қолданылған. Адрестік кеңістіктің ең соңында, F0000h...FFFFFFh (немесе E0000h...FFFFFFh) адрестерінде BIOS негізгі тұрақты сақтау құрылғысы орналасқан, ал C0000h – адресінен бастап BIOS – тың ТСК ұлғайтулары орналасқан, олар графикалық адаптерлер және дискілерді қадағалау үшін қолданылады. Жоғарғы жадының адрестік аймағының бір бөлігі графикалық адаптерлердегі видеобуфер адресациясы үшін арналған. Суретте көрсетілген видео буфердің орналасуы EGIA адаптерлері үшін арналған; ал басқа адаптерлер үшін ол басқа болуы мүмкін, (мысалы, қарапайым монохромды MDA адаптерлері үшін видеобуфер 4 Кбайт орын алады да B0000h адресі бойынша орналасады).

PC/AT компьютерлер құрамына стандартты (640Кбайт) жадыдан басқа кеңейтілген (ExtEndEd) жадының да кіруі мүмкін, оның максималды көлемі 80386/486 - 4 Гбайт процессорларын қолданған кездегі процессордың адрестік шинасына байланысты болады. Бұл жады бірінші мегабайт адрестік кеңістіктен кейін 100000h. адресі бойынша орналасады.

Дегенмен кеңейтілген жадының функциялануы "кеңейтілген жадының спецификациясына" бағынышты болғандықтан (ExtEndEd MEMory SpEciFicAtion, қысқаша XMS), бұл жадыны да әдетте XMS-жады деп атайды. Жоғарыда көсетілгендей, кеңейтілген жадыға қатынау қорғалған режимде орындалады, сондықтан MS DOS үшін, нақты режимде жұмыс жасайтын, кеңейтілген жады қолдануға берілмейді.

Бірінші 64Кбайт кеңейтілген жады, нақтырақ, 100000h-тан 10FFEFh-қа дейінгі 16 байт - 64Кбайт адрестерімен, арнайы жоғарғы жады (High MEMory ArEA, HMA) атына ие. Бұл аймақтың ерекшелігі, ол бірінші

мегабайт сыртында болса да, оған микропроцессор жұмысының нақты режимінде қатынауға болады.

**.EXE программаларының құрылымы және жадыда орналасуы**

MS-DOS-тың басқаруымен орындалатын программалар, келесі екі типтің біріне жатуы мүмкін, оларды программалық файлдардың кеңейтілуіне байланысты ажыратамыз: яғни, .COM немесе EXE программалар. .EXE типтегі программа кодында деректер және стек үшін арнайы сегменттер қарастырылған, ал .COM типіндегі програма бір ғана сегментпен жұмыс жасайды.

title	Программа	Типі	.EXE	
text	segment	'code'		
	assume	CS:text, DS:data		
Myproc	proc			
	move	AX, data		
	move	DS, AX		
	...			; программа тексті
	Myproc	Endp		
text	Ends			
data	segment			
	...			; деректерді анықтау
data		Ends		
stack	segment	stack 'stack'		
	dw	128dup (0)		
stack	Ends			
	End	Myproc		

Сонымен, .COM типіндегі программалардың өлшемі 64Кбайттан аспауы керек, ал .EXE типіндегі программаларға ондай шектеулер қойылмайды, өйткені онда кез-келген программалық сегменттер және деректер кіре алады.

Ассемблер тіліндегі .EXE типтегі программа құрылымы келесідегідей болады. Ескерте кететін жағдай, программа текстін пернетақтадан енгізген кезде үлкен әріптерді де және кіші әріптерді де қолдана беруге болады: транслятор келесі жолдарды бірдей қарайды MOV AX, DATA және Mov ax, data. Бірақ сәйкесінше кілттердің көмегімен трансляторды аттардағы үлкен және кіші әріптерді ескеретін етіп жасауға болады. Онда келесі программалық жолдардағы MYPROS proc және Myproc proc енді эквивалентті жолдар болып саналмайды.

Енді берілген программа құрылымын қарастырайық. TITLE операторы программаға тексттік тақырып жазу үшін қолданылады, ол трансляция листингі кезіндегі барлық парақтарға шығарылады. Программа үш сегменттен тұрады: команда сегменті, немесе өзгеріске алынған text аты бар программалық сегмент, data аты бар деректер сегменті және stack аты бар стек сегменті (бұл аттар да өзгеріске таңдалуы мүмкін).

Әрбір сегмент SEGMENT операторымен ашылып және ENDS операторымен жабылады. Бұл екі оператордың алдында міндетті түрде сегмент аты болуы керек. Ал сегменттердің орналасу тәртібі көп жағдайда маңызды емес.

Командалар сегментінің түсініктемелер жолында апострофта орналасқан 'CODE' сөзі, сегмент класының "программалық" екендігін көрсетеді. Сегменттер класы копоновщикпен тексеріледі және жүктемелегіш модулінің компоновкасы кезінде қолданылады: бір класқа жататын сегменттер, жадыға бірінен соң бірі жүктемеленеді.

Бір команда сегменті және бір деректер сегменті бар, қарапайым программалар үшін, бұл процедураның мәні жоқ, бірақ LINK копоновщикінің дұрыс жұмыс істеуі үшін және командалар сегментінің түсініктемесі кезінде CodeView аудармалағышы үшін 'CODE' класын көрсеткен дұрыс.

Командалар сегментінің тексті ASSUME операторынан басталады, ол трансляторға сегменттік регистрлер және олармен адресацияланатын сегменттерді сәйкестендіруге мүмкіндік береді. CS анықтағышы: text трансляторға берілген сегмент программалық екендігін және ол CS сегменттік регистрдің көмегімен адрестелетіндігін көрсетеді. DS анықтағышы: data DS сегменттік регистрін data сегментіне бекітеді, ол арқылы DS регистрін көрсетпей ақ data сегментінде көрсетілген айнымалыларға сілтеме жасауға болады. Бірақ бұл кезде ассемблер ол деректердің data сегментінде бар – жоқ екендігін тексеріп отырады.

Сондай – ақ программа процедуралардан тұруы мүмкін. Программаны процедураларға бөлу маңызды емес, бірақ ішкі программаларда басқаруды ауыстыру және басқа программалық модулдерге ауысу кезінде өте қолайлы болып есептелінеді. Қаралып отырған мысалдағы команда сегменті Murgos процедурасынан тұрады, ол PROC операторымен ашылады және ENDP операторымен жабылады. Бұл екі оператордың алдында міндетті түрде процедура аты болуы керек.

Программаның бірінші жолдарында DS регистрі инициализацияланады – онда деректер сегментінің сегмент адресі орналасады. Сегменттік регистрлерге тікелей мән беруге болмағандықтан, "алдау пункті" ретінде AX регистрі қолданылады. Бұл жұмыстардан кейін, яғни DS регистрінің инициализациясынан соң, программа деректер сегментінде орналасқан деректерді қолдана алады.

Деректер сегменті программада қолданылған барлық айнымалылар жайындағы түсініктемелерден тұрады. Деректер түсініктемелерінің орналасу реті төменде қаралады.

Стек сегментінің түсініктемелер жолында сегмент класы - 'STACK', сондай-ақ біріктіру типі – STACK көрсетілуі керек. Біріктіру типі компоновщикке, әртүрлі модульдегі бірдей атты сегменттерді бірін-бірі жабу (COMMON біріктіру типі) немесе бірін-біріне қосылу (стек сегменттері үшін STACK біріктіру типі немесе PUBLIC барлық қалғандары үшін) арқылы қалай біріктіруге болатындығын көрсетеді. Бірақ бірмодулді программалар үшін біріктіру типі маңызды емес, стек сегменті үшін STACK міндетті түрде берілетін біріктіру типі, өйткені бұл жағдайда программаны жүктемелеген кезде автоматты түрде SS регистрі инициализацияланады (стек сегментінің бастапқы адресі) және SP (стек сегментінің соңының ығысуы). Берілген мысалда стек үшін 128 сөз жады бөлінген. Стек сегментінің класын көрсету міндетті емес, бірақ программалаушыға программа сегментінің жадыда орналасу реті маңызды болса, онда сегменттерге бір-бірінен айырмашылығы болатын класс аттарын берген дұрыс (бір сегменттің класы болмауы да мүмкін). Бұл жағдайда компоновщик сегменттерді жадыға программада орналасу реті бойынша орналастырады.

Программа тексті трансляцияны аяқтатын END директивасымен аяқталады. Бұл директива операнды ретінде бас процедураның кіріс нүктесі алынады.

Программа жүктемеленген кезде сегменттер жадыға орналастырылады, ол 2.1 суретте көрсетілген

Жадыдағы программа программалық сегмент префиксінен басталады (program Segment Prefix, ПСП), ол жүйемен толтырылады.

Әдетте ПСП өлшемі 256 байт болады, ол программаның орындалу процесінде жүйемен қолданылатын кестелер және деректер полесінен тұрады. Бұл полелердің кейбіреулері төменде көрсетілген. ПСП-ден кейін программа сегменттері орналасады. Сегменттік регистрлер автоматты түрде келесідегідей инициализацияланады: ES және DS ПСП басын көрсетеді (ол олардың құрамын сақтауға мүмкіндік береді, одан кейін программада ПСП-ге қатынауға болады), CS- командалар сегментінің басы, ал SS- стек сегментінің басы. IP команда көрсеткішіне программаға кіру нүктесінің салыстырмалы адресі жүктмеленеді (END директивасының операндынан), ал SP стек көрсеткішіне – стек сегментінің соңының ығысуы жүктемеленеді. Сонымен программаның жадыға жүктемеленуінен кейін барлық сегменттер адрестелген болады, деректер сегментінен басқа. DS регистрінің программаның бірінші жолдарында пайда болуы бұл сегментті де адрестей алады.

ES,DS----->	ПСП 256 байт	
CS----->	Команда сегменті	<-----IP
	Деректер сегменті	

SS→	Стек сегменті	
		←-----SP
2 сурет. .EXE программалардың жадыда орналасуы		

### .COM программаларының құрылымы және жадыда орналасуы

Жоғарыда айтылғандай, .COM типтегі программалардың .EXE типтегі программалардан айырмашылығы, онда тек бір сегмент қана болады, ол программа компоненттерінің барлығынан тұрады: ПСП, программалық код (яғни машиналық кодқа ауысқан программалық жолдар), стек және деректер. Ассемблер тіліндегі .COM типіндегі программа келесідегідей түрде беріледі:

title	Программа	Типі	.COM	
text	segment	'code'		
	assume	CS: text,	DS:text	
	org	100h		
Myproc	proc			
	...			;Программа тексті
Myproc	Endp			
	...			;Деректерді анықтау
text	Ends			
	End	Myproc		

Программа бір ғана text сегментінен тұрады, оған 'CODE' класы меншіктелген. ASSUME операторында CS және DS сегменттік регистрлері көрсетілген. ORG 100h операторы ПСП үшін 256 байтты резервтейді. ПСП-ді бұрынғыдай жүйе жасайды, бірақ оған қалдырылатын орынды сегменттің басында программалаушы беріп кету керек. Программада DS сегменттік регистрін және басқа сегменттік регистрлерді инициализациялаудың керегі жоқ, өйткені оны жүйе өзі жасайды. Деректерді программалық процедурадан кейін (суретте көрсетілгендей) немесе оның ішінде, не одан бұрын беріп кетуге болады. Бірақ ескере кететін жағдай, .COM типіндегі программаларды жүктемелеу кезінде IP регистрі әруақытта 100 h санымен жүктемеленеді, сонымен ORG 100h операторынан кейін міндетті түрде бірінші орындалатын программа жолы орналасуы керек. Егер деректер программа басында орналасқан болса, онда оның алдына нақты кіру нүктесіне өтетін операторды орналастырған дұрыс, мысалы: JMP Entry.

3-суретте .COM типтегі программаның жадыда орналасуы көрсетілген. Программа жүктемеленгеннен кейін барлық сегменттік регистрлер бір ғана

сегменттің басын көрсетеді, яғни ПСП-дің басын көрсетеді десек те болады. Стек көрсеткіші автоматты түрде FFFh санымен инициализацияланады. Сонымен, программаның нақты өлшеміне қарамастан оған 64 Кбайт адрестік кеңістік бөлінеді, ал қалған төменгі бөлікті стек пайдаланады. Дегенмен стектің жоғарғы шегарасы анықталмағандықтан және программның стекті пайдаланғанына байланысты болғандықтан, программаның төменгі бөлігінің жойылып кетуінен сақ болған жөн. Бірақ, мұндай жағдай .EXE типтегі программаларда да кездеседі.

CS, DS, ES, SS →	ПСП 256 байт	
	Программа және деректер	← IP
	Стек	
		← SP = FFFh

3-сурет. .COM программаларының жадыда орналасуы

#### Бақылау сұрақтары:

1. Кез-келген жады ұяшығының 20-разрядтық адресі қалай есептелінеді?
2. 80386 және 80486 МП деректер және адресер шинасының разрядтылығы?
3. .COM және .EXE программаларының форматтарының айырмашылығы?

### 2.1 Микропроцессордың пайдаланушы регистрлері

Микропроцессордың программалық моделі программалаушы қолдануға арналған 32 регистрдан тұрады. Берілген регистрлерді екі үлкен топқа бөлуге болады: 16 пайдаланушылар регистрі және 16 жүйелік регистрлер.

Ассемблер тілінде жазылған программаларда регистрлердің қолданылуы өте интенсивті түрде жүреді. Көптеген регистрлердің анықталған өз функционалдық қолданылуы бар.

#### Пайдаланушылар регистрлері

Бұл регистрлерге жатады (сурет 1):

- Сегіз 32-биттік регистрлер, олар программалаушымен адресер мен деректерді сақтау үшін қолданылады (оларды сондай-ақ жалпы міндетті регистрлер деп те атайды (ЖМР)):

- eax/ax/ah/al;
- ebx/bx/bh/bl;
- edx/dx/dh/dl;
- ecx/cx/ch/cl;

- ebp/bp;
- esi/si;
- edi/di;
- esp/sp.
- Алты сегмент регистрі cs, ds, ss, es, fs, gs;
- Басқару және күйлер регистрі:
- белгілер регистрі eflags/flags;
- команда көрсеткіші регистрі eip/ip.

i8086 микропроцессорларынан бастап, Intel фирмасынан шыққан барлық кіші 16-разрядтық моделдерге арнап жазылған программалардың дұрыс жұмыс істеуі үшін, үлкен 32-разрядты регистрлерді бөліп бір бөлігін қолдануға болады, 32-разрядты регистрлер е әріпімен басталады (*Extended*).

### Жалпы міндетті регистрлер

Бұл топтағы барлық регистрлер өздерінің «кіші» бөліктеріне қатынас жасауға мүмкіндік береді. Өз еркінше адрестеу жасау үшін бұл регистрлердің 16 және 8-биттік кіші бөліктерін ғана қолдануға болады. Ал бұл регистрлердің үлкен 16 биті өз еркінше объек, қолдануға болмайды.

Жалпы міндетті регистрлерге келесі регистрлерді жатқызуға болады:

- eax/ax/ah/al (Accumulator register) — аккумулятор.
- ebx/bx/bh/bl (Base register) — базалық регистр.
- ecx/cx/ch/cl (Count register) — регистр-санауыш.
- edx/dx/dh/dl (Data register) — деректер регистрі.

EAX, EBX, ECX, EDX регистрлері кез-келген мақсаттарда қолданылы береді — әртүрлі операциялардың нәтижелерін және аргументтерін, деректерді уақытша сақтау.

Келесі екі регистр тізбекті операцияларды орындау үшін керек, яғни 8,16 немесе 32 битті ұзындығы бар элементтер тізбегін біртіндеп өңдеу жүргізуге мүмкіндік беретін регистрлер:

esi/si (Source Index register) — таратқыш индексі. Тізбекті операциялардағы бұл регистр ағымдағы тізбек — таратқыш элементінің адресінен тұрады;

edi/di (Destination Index register) — қабылдағыш индексі (қабылдап алушы).

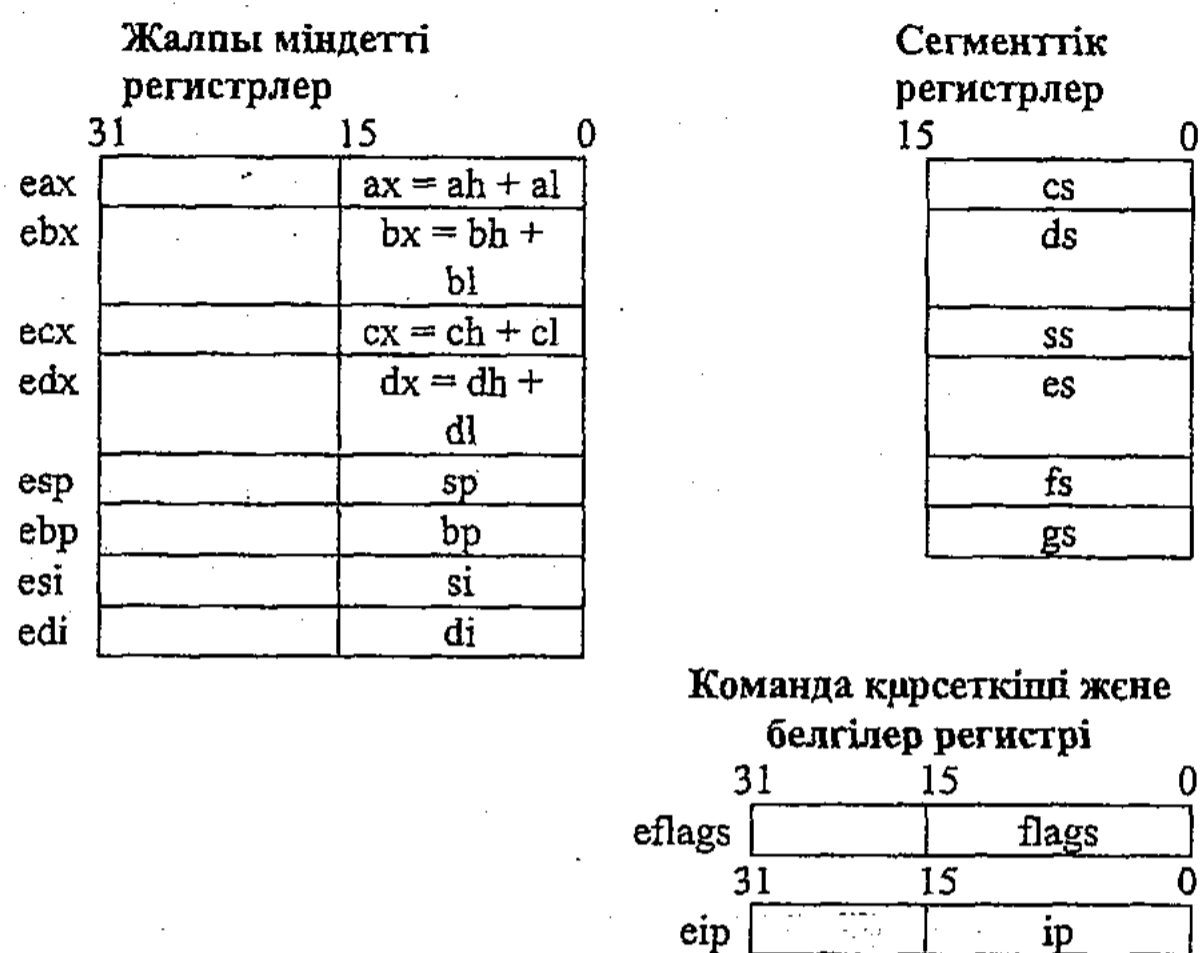
Тізбекті операциялардағы бұл регистр ағымдағы тізбек — қабылдағыш элементінің адресінен тұрады.

Микропроцессордың архитектурасында программалық — аппараттық деңгейде стек атты деректер құрылымы пайдаланылады. Стекпен жұмыс істеу үшін микропроцессор командалар жүйесінде арнайы командалар бар, ал микропроцессордың программалық моделі үшін арнайы регистрлер пайдаланылады:

- esp/sp (Stack Pointer register) - стек көрсеткіш регистрі.
- Ағымдағы стек сегментіндегі стек төбесінің көрсеткішінен тұрады.
- ebp/bp (Base Pointer register) - стек база кадрының көрсеткіш регистрі.

Стек ішіндегі деректерге қатынауды өз еркінше ұйымдастыру үшін арналған.

Көптеген регистрлер программалау кезінде кез-келген түрде операндаларды сақтау үшін қолданылады.



4-сурет. i486 және Pentium микропроцессорларының пайдаланушылар регистрлері

### Сегменттік регистрлер

Микропроцессордың программалық моделінде алты сегменттік регистрлер бар: cs, ss, ds, es, gs, fs. Микропроцессор аппараттық түрде программа құрылымын үш түрлі бөлікте ұйымдастырады, оларды сегменттер деп атаймыз. Сәйкесінше, мұндай жадыны ұйымдастыру сегменттік деп аталады.

Сегменттік регистрлер нақты уақыт ішіндегі программа қатынасатын сегменттерді көрсету үшін қолданылады. Нақтылай айтатын болсақ бұл регистрлерде сәйкесінше сегмент орналасқан жады адрестері орналасады. Машиналық команданың өңделуінің логикасының құрылымы, команданы тандау кезінде, стекке немесе программа деректеріне қатнау кезінде анықталған сегменттік регистрлердің адрестерін қолданады. Микропроцессор келесі түрдегі сегменттерді қолдайды:

1. **Код сегменті.** Программа командаларынан тұрады. Бұл сегментке қатынау үшін cs регистрі қолданылады (code segment register) – кодтың - сегменттік регистрі. Микропроцессор қатынай алатын, машиналық командалардан құрылған сегмент адресінен тұрады (яғни бұл командалар микропроцессор конвейеріне жүктеледі).

2. **Деректер сегменті.** Программамен өңделетін деректерден тұрады. Бұл сегментке қатынау үшін ds регистрі қолданылады (data segment register) – деректердің сегменттік регистрі, мұнда ағымдағы программаның деректер сегментінің адресі сақталады.