

1-дәріс. Компьютерлік жүйелерге шолу

- 1.1. Компьютерлік жүйелердің негізгі
- 1.2. элементтері
- 1.2. Микропроцессорлардың эволюциясы
- 1.3. Командаларды орындау
- 1.4. Үзілістер
- 1.5. Сақтау құрылғыларының иерархиясы
- 1.6. Көп процессорлы және көп ядролы жүйелерді ұйымдастыру

Операциялық жүйе бір немесе бірнеше процессорларды қамтитын аппараттық ресурстарға жүгіну кезінде пайдаланушыларға қызмет көрсетеді. Сонымен қатар, ол екінші жад пен енгізу-шығару құрылғыларын басқарады. Сондықтан, операциялық жүйелерді зерттеуге кіріспес бұрын, олар жұмыс істейтін компьютерлік жүйелер туралы біраз түсінік алу маңызды.

1.1. НЕГІЗГІ ЭЛЕМЕНТТЕР

Жоғарғы деңгейде компьютер процессордан, жадтан және енгізу-шығару құрылғыларынан тұрады; әр компонент бір немесе бірнеше модульмен ұсынылған. Компьютер бағдарламаларды орындаудан тұратын өзінің негізгі мақсатын орындай алуы үшін әртүрлі компоненттер өзара әрекеттесе алуы керек. Компьютердің төрт құрылымдық компонентін ажыратуға болады.

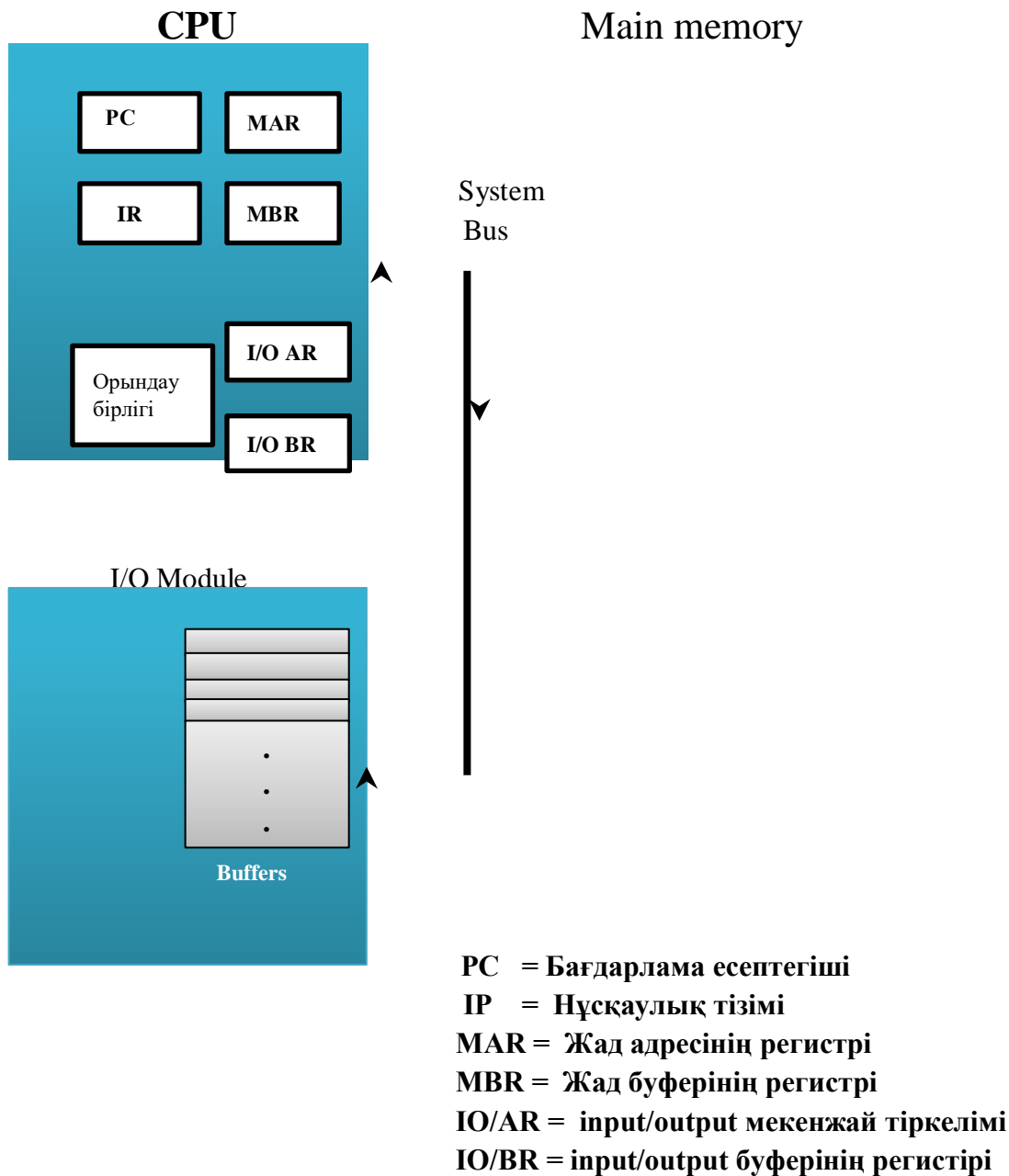
* **Процессор.** Компьютердің барлық әрекеттерін басқарады, сонымен қатар деректерді өңдеу функциясын орындайды. Егер жүйеде тек бір процессор болса, оны көбінесе орталық процессор (Central Processing Unit – CPU) деп атайды.

* **Негізгі жад.** Онда деректер мен бағдарламалар сақталады. Әдетте, бұл жад уақытша, яғни компьютерді өшірген кезде оның мазмұны жоғалады. Оны көбінесе нақты, жедел немесе бастапқы жад деп атайды.

* **Енгізу-шығару құрылғылары.** Компьютер мен әртүрлі перифериялық құрылғылардан тұратын сыртқы орта арасында деректерді беру үшін қызмет етеді, олардың ішінде екінші жад, байланыс жабдықтары және терминалдар бар.

* **Жүйелік шина.** Процессор, негізгі жад және енгізу-шығару құрылғылары арасындағы өзара әрекеттесуді қамтамасыз ететін белгілі бір құрылымдар мен механизмдер.

Аталған компоненттер 1.1-суретте көрсетілген. Процессордың функцияларының бірі-деректерді жадпен бөлісу. Ол үшін ол әдетте екі ішкі (процессорға қатысты) регистрді қолданады: Жад мекен-жайының регистрі (memory address register — MAR), онда оқу-жазу операциясы орындалатын жад ұяшығының мекен — жайы жазылады және жад буферінің регистрі (memory buffer register-MBR), онда жазуға арналған деректер енгізіледі жадында немесе одан оқылғандар жазылады. Сол сияқты, енгізу-шығару құрылғысының нөмірі енгізу-шығару мекенжайының регистрінде (I/O address register — I/O AR) беріледі. Енгізу-шығару буферінің регистрі (I/O buffer register — I / O BR) енгізу— шығару құрылғылары мен процессор арасында деректер алмасуға қызмет етеді.



1.1-сурет. Компьютер компоненттері: жалпы құрылым

Жад модулі көптеген нөмірленген ұяшықтардан тұрады. Әр ұяшыққа екілік санды жазуға болады, ол команда ретінде де, деректер ретінде де түсіндіріледі - енгізу-шығару модулі сыртқы құрылғылардан деректерді процессор мен жадқа да, кері бағытта да тасымалдауға қызмет етеді. Деректерді уақытша сақтау үшін оның ішкі буферлері бар.

1.2 Микропроцессорлардың эволюциясы

Процессор бір чипте болған микропроцессорлар үнемі дамып отырады. Бірақ микропроцессорлар тек жылдам және қол жетімді жалпы мақсаттағы процессорларға айналды; олар қазір мультипроцессорлар - әр чипте бірнеше процессорлар бар (ядролар деп аталады). Мұндай ядроның әрқайсысында үлкен көлемді жад кэштерінің бірнеше деңгейі бар, ал бірнеше логикалық процессорлар әр ядроның атқарушы модульдерін бөліседі. Процессорлар есептеудің көптеген түрлері үшін өте жақсы өнімділікті қамтамасыз еткенімен,

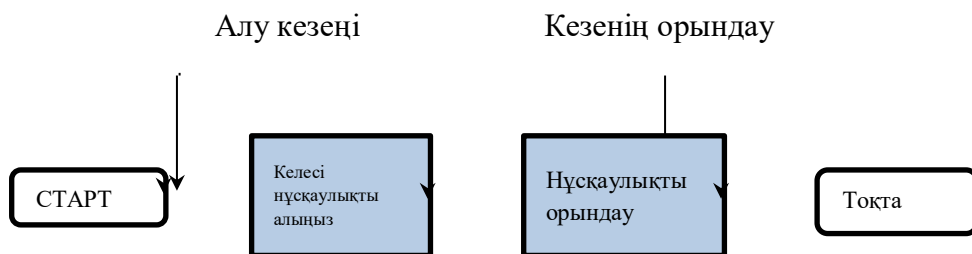
сандық есептеулерге сұраныс артып келеді. GPU (graphical Processing Units-GPU) бір команданың (Single - Instruction Multiple Data-SIMD) көптеген деректерді өңдеу мүмкіндігін пайдалана отырып, деректер массивтерінде тиімді есептеулерді қамтамасыз етеді. Сонымен қатар, процессорлар деректер массивтерімен жұмыс істеу үшін көбірек мүмкіндіктерге ие болады. X86 және AMD64 отбасыларының процессорларының архитектурасы барған сайын қуатты векторлық модульдерді біріктіреді. Аудио немесе бейне сияқты ағындық сигналдармен жұмыс істеуге арналған сандық сигналдарды (Digital Signal Processors - DSP) өңдеуге арналған процессорлар бар. Портативті құрылғыларға қойылатын талаптарды қанағаттандыру үшін классикалық микропроцессор чиптегі жүйелерге жол береді, онда бір чипте тек процессор мен кэш жады ғана емес, сонымен қатар DSP, GPU, енгізу-шығару құрылғылары (мысалы, радио және кодектер), сондай-ақ негізгі жады бар.

1.3. Командаларды орындау

Процессор орындайтын бағдарлама жадта сақталған командалар жиынтығынан тұрады. Қарапайым түрде командаларды өңдеу екі кезеңде жүреді: процессор жадтан оқиды (таңдайды), содан кейін келесі пәрменді іске қосады. Бағдарламаның орындалуы команданы іріктеу процесін қайталауға және оны орындауға дейін азаяды. Бір команданы орындау үшін бірнеше операция қажет болуы мүмкін; олардың саны команданың табиғатымен анықталады.

Бір команданы іске асыру үшін қажет әрекеттер жиынтығы оның циклі деп аталады.

1.2-суретте іріктеу кезеңі мен орындау кезеңдерін қамтитын жеңілдетілген схемада процессордың командаларды өңдеу процесі көрсетілген.



1.2-сурет. Бағдарламаны орындаудың негізгі циклі

Шығарылған командалар командалар тізіміне (IR) жүктеледі. Пәрмен процессорға қандай әрекеттерді орындау керектігін көрсететін биттер тізбегінен тұрады. Процессор команданы түсіндіреді және қажетті әрекеттерді орындайды. Барлық әрекеттерді төрт санатқа бөлуге болады.

* **Процессор-жад.** Деректер процессордан жадқа немесе кері жіберіледі.

* **Процессор-енгізу-шығару құрылғылары.** Процессордан алынған деректер перифериялық құрылғыға енгізу-шығару құрылғысы арқылы келеді. Кері процесс те мүмкін.

* **Деректерді өңдеу.** Процессор деректермен әртүрлі арифметикалық немесе логикалық амалдарды орындайды.

* **Басқару.** Пәрмен командалардың орындалу ретін өзгертуді орната алады.

1.4 Үзілістер

Барлық компьютерлерде әртүрлі құрылғылар (енгізу-шығару, жад) процессордың қалыпты жұмысын тоқтата алатын механизм бар. Үзілістердің жалпы қабылданған негізгі

сыныптары 1.1-кестеде келтірілген.

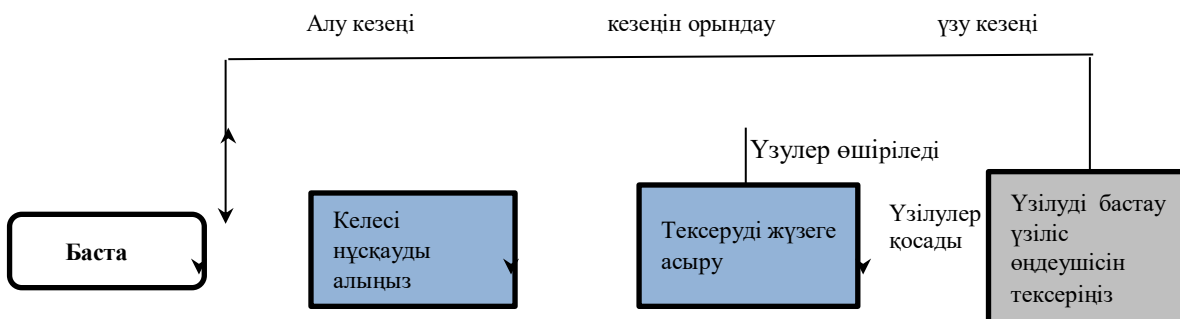
Кесте 1.1. Үзіліс сыныптары

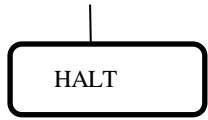
Бағдарламалық үзіліс	Командаларды орындау нәтижесінде пайда болатын кейбір жағдайларда жасалады. Мұндай жағдайлар арифметикалық толып кету, нөлге бөлу, қате пәрменді орындауға тырысу және пайдаланушыға кіруге тыйым салынған жад аймағына сілтеме болуы мүмкін
Таймер үзілісі	Процессор таймер арқылы жасалады. Бұл үзіліс амалдық жүйеге белгілі бір уақыт аралығында өзінің кейбір функцияларын мезгіл-мезгіл орындауға мүмкіндік береді
Енгізу-шығару үзілісі	Енгізу-шығару контроллері арқылы жасалады. Операцияның қалыпты аяқталуы немесе қателіктердің болуы туралы сигнал береді
Аппараттық үзіліс	Желідегі кернеудің төмендеуі немесе жад паритетін басқару қатесі сияқты ақаулар мен төтенше жағдайлар кезінде пайда болады

Үзілістер негізінен жұмыс тиімділігін арттыруға арналған. Мысалы, енгізу-шығару құрылғыларының көпшілігі процессорға қарағанда әлдеқайда баяу жұмыс істейді. Процессор деректерді принтерге 1.2-суретте көрсетілген схема бойынша жібереді делік. Әр операциядан кейін процессор кідіруге мәжбүр болады және принтер деректерді қабылдағанша күтеді. Бұл кідірістің ұзақтығы жад қоңыраулары қатысатын топ циклінің ұзақтығынан мыңдаған және миллиондаған есе көп болуы мүмкін. Процессорды мұндай пайдалану тиімсіз екені анық.

Енгізу-шығару операцияларын орындау кезіндегі үзілістердің арқасында процессор басқа командаларды өңдеумен айналысуы мүмкін. Пайдаланушы бағдарламасы тұрғысынан үзілістер әдеттегі орындау ретін бұзады. Үзілісті өңдеу аяқталғаннан кейін жұмыс қайта басталады. Пайдаланушы бағдарламасын тоқтата тұру және оның жұмысын үзілген жерден қалпына келтіру үшін процессор мен операциялық жүйе жауап береді.

Үзілісті бағдарламамен үйлестіру үшін командалық циклге үзіліс циклі қосылады (1.3-сурет). Үзіліс циклінде процессор болған үзілістерді көрсететін үзіліс сигналдарының болуын тексереді. Үзіліс түскен кезде процессор ағымдағы бағдарламамен жұмысын тоқтатады және үзіліс өңдегішін орындайды. Үзіліс өңдегіштері әдетте операциялық жүйенің бөлігі болып табылады. Әдетте, бұл бағдарламалар үзілістің сипатын анықтайды және қажетті әрекеттерді орындайды.

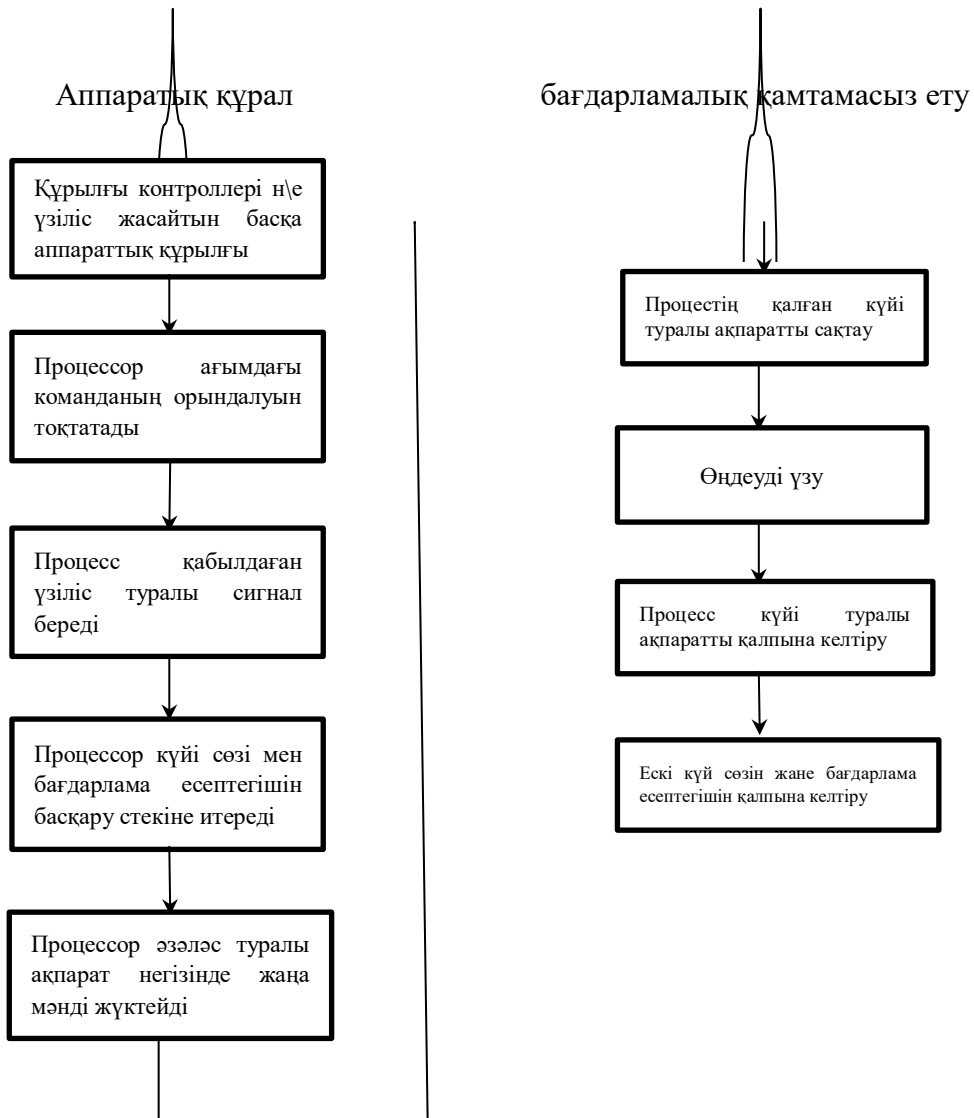




1.3-сурет. Үзіліс командасының циклі

Үзілістерді өңдеу

Үзіліс аппараттық және бағдарламалық жасақтамада болатын бірқатар оқиғаларды тудырады. 1.4-суретте осы оқиғалардың типтік тізбегі көрсетілген. Енгізу-шығару құрылғысы аяқталғаннан кейін келесілер орын алады.



1.4-сурет. Қарапайым үзілісті өңдеу

1.5. Жады иерархиясы

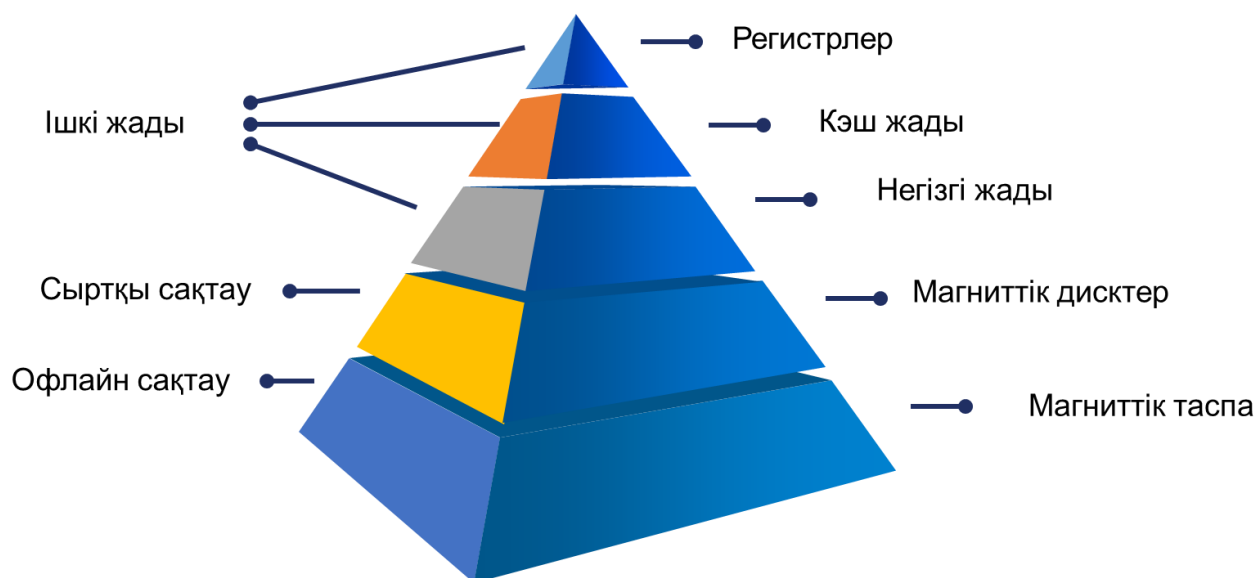
1.5.1 Компьютер жадының конфигурациясы

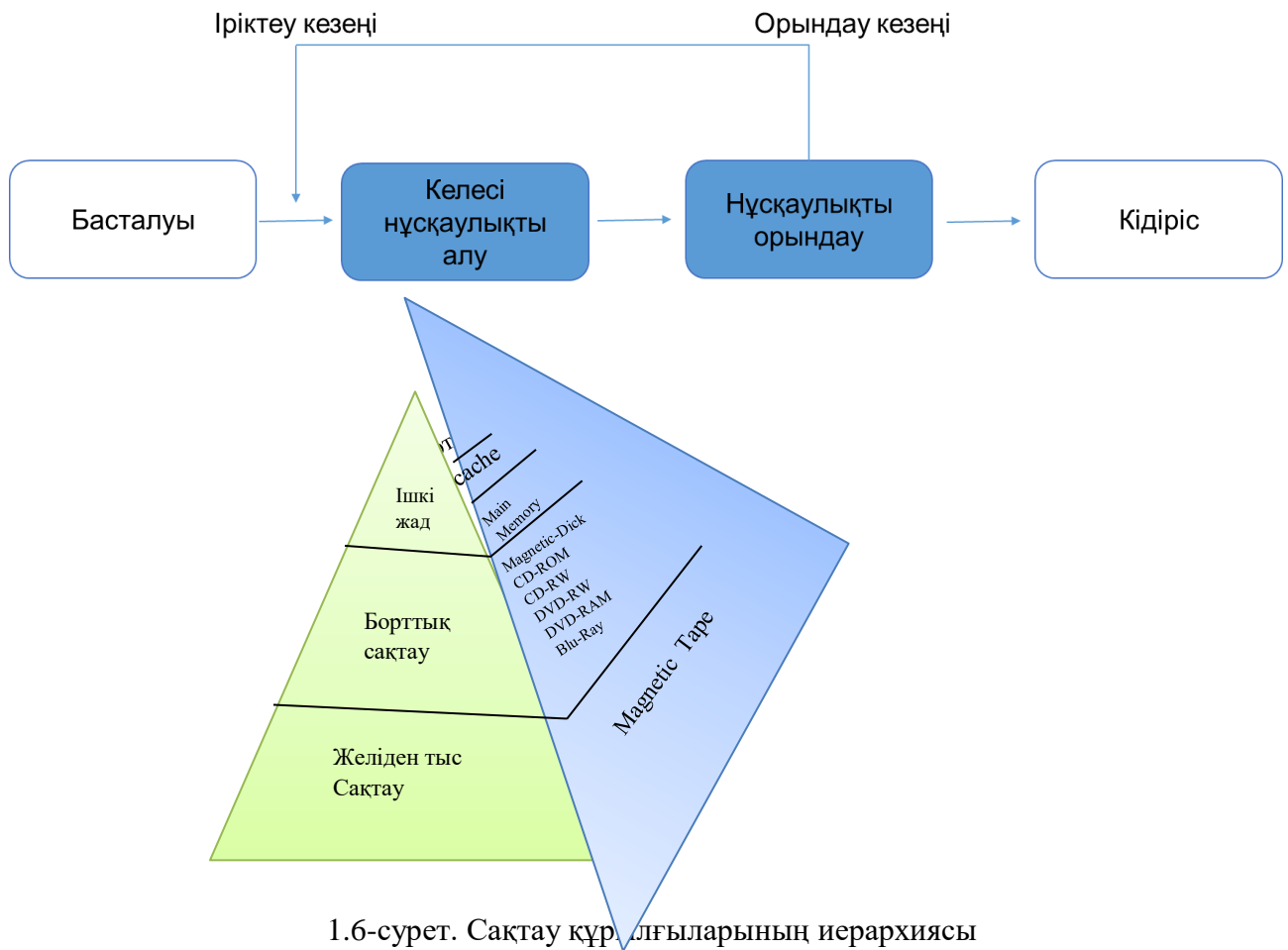
Компьютер жадының конфигурациясы негізінен үш параметрмен анықталады: көлемі, жылдамдығы, құны. Сақтау құрылғыларын өндіру технологияларын дамытудың кез-келген кезеңінде келесі тұрақты қатынастар жұмыс істейді.

- * Кіру уақыты неғұрлым аз болса, әр бит соғұрлым қымбат болады.
- * Сыйымдылық неғұрлым жоғары болса, бит құны соғұрлым төмен болады.
- * Сыйымдылық неғұрлым жоғары болса, қол жеткізу уақыты соғұрлым көп болады.

Бұл жағдайдан шығудың жолын табу үшін жеке жад компонентіне немесе технологияға емес, есте сақтау құрылғыларына иерархия құру қажет. 1.6-суретте сақтау құрылғыларының типтік иерархиясы көрсетілген. Оның негізіне түсу кезінде келесілер орын алады.

1. Биттің құны төмендейді.
2. Сыйымдылығы артады.
3. Қол жеткізу уақыты артады.
4. Процессордың жадқа кіру жиілігі төмендейді.





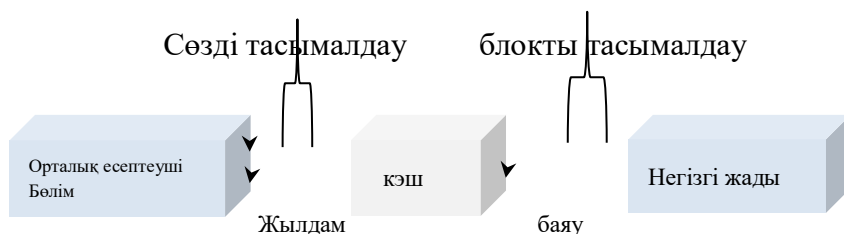
1.6-сурет. Сақтау құрылғыларының иерархиясы

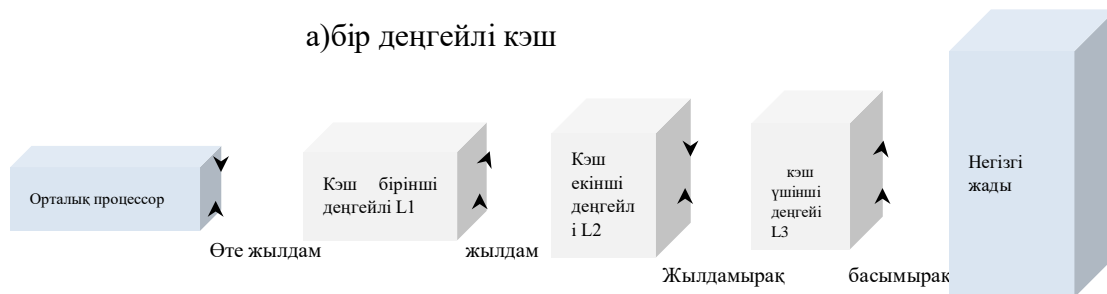
Сыйымдылығы төмен және өнімділігі жоғары қымбат құрылғыларға сыйымдылығы жоғары, арзан, бірақ өнімділігі төмен құрылғылар қосылады. 4-шартты сақтаудың негізі өтініштерді оқшаулау принципі деп аталатын принцип болып табылады [DEN68]. Бағдарламаны орындау кезінде процессор жүгінетін мекен-жайлар топтарға жиналуға бейім. Цикл немесе ішкі кезек пайда болғаннан кейін, процессор тек қайталанатын командалар жиынтығына жүгінеді. Уақыт өте келе кейбір кластерлер басқалармен ауыстырылады, бірақ аз уақыт ішінде процессор негізінен бекітілген жад кластерлерімен жұмыс істейді.

1.5.2 Кэш

Негізгі жадты кеңейту-бұл сыйымдылығы аз, бірақ жылдамырақ кэш. Бұл құрылғы негізгі жад пен процессор регистрлері арасында жылжу кезінде деректердің аралық қоймасы ретінде әрекет етеді және өнімділікті арттыруға мүмкіндік береді. Кэш амалдық жүйеге көрінбейді, бірақ ол жадпен байланысты аппараттық құралдармен өзара әрекеттеседі.

Кэш жадқа қол жеткізу жылдамдығын мүмкіндігінше жақындатуға және сонымен бірге жартылай өткізгіш жадының арзан түрлерінің бағасына үлкен көлемдегі жадты қамтамасыз етуге арналған. Бұл тұжырымдама 1.7а-суретте көрсетілген.





б) үш деңгейлі кэш ұйымдастыру

1.7-сурет. Кэш және негізгі жады

Кэш негізгі жадының көшірмесін сақтайды. Процессор жадыдан байтты немесе сөзді оқуға тырысқанда, бұл сөздің кэште бар-жоғын тексеру жүргізіледі. Егер бар болса, бұл байт немесе сөз процессорға беріледі. Егер ол жоқ болса, белгілі бір адресі бар сөздерден тұратын негізгі жады блогы кэшке оқылады, содан кейін қажетті байт немесе сөз процессорға беріледі. Қажетті сөздердің бірін қамтитын деректер блогын кэшке оқу кезінде өтініштердің локализациясына байланысты деректерге кейінгі өтініштер де осы блоктағы сөздерге орындалатын болады.

1.7-суретте бірнеше деңгейден тұратын кэш көрсетілген. L2 кэші L1 кэшінен баяу және үлкенірек, ал L3 кэші L2 кэшінен баяу және үлкенірек. 1.8-суретте негізгі жады пен кэш құрылымы көрсетілген. Негізгі жады $2n$ адресік сөзден тұрады, олардың әрқайсысы өзінің бірегей n -биттік мекен-жайымен сипатталады. Дисплей мақсаттары үшін барлық жады әрқайсысы K сөзден тұратын белгіленген ұзындықтағы блоктардың белгілі бір санынан тұрады деп болжанады.

Осылайша, барлығы $m = 2n/k$ блоктары бар. Кэш слоттардан тұрады (сызықтар деп те аталады). Бұл жағдайда слоттар саны блоктар санынан әлдеқайда аз ($бирге \ll M$). Негізгі жады блоктарының кейбір жиынтығы кэш слоттарында сақталады. Егер сіз жадыдан кэште жоқ блоктың сөзін оқуыңыз керек болса, онда бұл блок кэш слоттарының біріне беріледі. Процессор оқылатын сөздің мекен-жайын жасайды. Егер бұл сөз кэште сақталса, ол процессорға беріледі. Әйтпесе, бұл сөзді қамтитын блок кэшке жүктеледі және сөз процессорға беріледі.

1.6. КӨП ПРОЦЕССОРЛЫ ЖӘНЕ КӨП ЯДРОЛЫ ЖҮЙЕЛЕРДІ ҰЙЫМДАСТЫРУ

Дәстүрлі түрде компьютер дәйекті әрекеттерді орындауға арналған машина ретінде қарастырылады. Көптеген бағдарламалау тілдерінде алгоритм дәйекті нұсқаулар түрінде беріледі; бағдарлама жұмыс істеп тұрған кезде процессор машиналық командаларды бірінен соң бірін орындайды. Әр команда операциялар тізбегі түрінде ұсынылады (команданы іріктеу, операндтарды іріктеу, операцияны орындау, нәтижелерді сақтау).

Компьютерге деген мұндай көзқарас ешқашан толық шындыққа сәйкес келмеді. Микрооперация деңгейінде бір уақытта бірнеше басқару сигналдары жасалады. Кем дегенде іріктеу және орындау операцияларын бір уақытта орындауға мүмкіндік беретін командаларды конвейерлік өңдеу бұрыннан қолданылып келеді. Жоғарыда келтірілген мысалдардың екеуі де функциялардың параллель орындалуының үлгілері болып табылады.

Компьютерлік технологиялар дамып, аппараттық құралдардың құны төмендеген сайын, компьютер әзірлеушілері параллелизмді жүзеге асырудың көбірек мүмкіндіктерін тапты. Бұл әдетте өнімділікті арттыру үшін, ал кейбір жағдайларда сенімділікті арттыру үшін жасалды. Бұл кітап мультипроцессорлық жүйелерде параллелизмді қамтамасыз етудің ең

танымал үш тәсілін зерттейді: симметриялы мультипроцессорлық (symmetric multiprocessor - SMP), көп ядролы компьютерлер және кластерлер.

1.6.1 Симметриялы көппроцессорлық

Анықтама. SMP келесі сипаттамалары бар оқшауланған компьютерлік жүйе ретінде анықталуы мүмкін.

1. Салыстырмалы мүмкіндіктері бар кем дегенде екі ұқсас процессор бар.
2. Бұл процессорлар бірдей негізгі жад пен енгізу-шығару құрылғыларын пайдаланады және автобуспен немесе басқа ішкі байланыс схемасымен қосылады, осылайша жадқа кіру уақыты әр процессор үшін бірдей болады.

3. Барлық процессорлар енгізу-шығару құрылғыларына бірдей арналар арқылы немесе бір құрылғыға жол беретін әртүрлі арналар арқылы ортақ қол жеткізе алады.

4. Барлық процессорлар бірдей функцияларды орындай алады (демек, термин симметриялы).

5. Жүйе процессорлар мен олардың бағдарламалары арасындағы тапсырмалар, есептер, файлдар және деректер элементтері деңгейінде өзара әрекеттесуді қамтамасыз ететін біріктірілген операциялық жүйемен басқарылады.

5-тармақ кластер сияқты әлсіз байланысқан мультипроцессорлық жүйеден айырмашылықты көрсетеді. Соңғысында өзара әрекеттесудің физикалық бірлігі әдетте хабарлама немесе толық файл болып табылады. SMP-де өзара әрекеттесу деңгейі жеке деректер элементтері болуы мүмкін және процестер арасында өзара әрекеттесудің жоғары дәрежесі болуы мүмкін.

SMP ұйымы бір процессорлы ұйымға қарағанда бірқатар әлеуетті артықшылықтарға ие, олар төменде көрсетілген:

Өнімділік. Егер компьютер орындауы керек жұмысты оның кейбір бөліктерін параллель орындауға болатындай етіп ұйымдастыруға болатын болса, онда көп процессорлы жүйе бір типтегі бір процессорға қарағанда үлкен өнімділік береді.

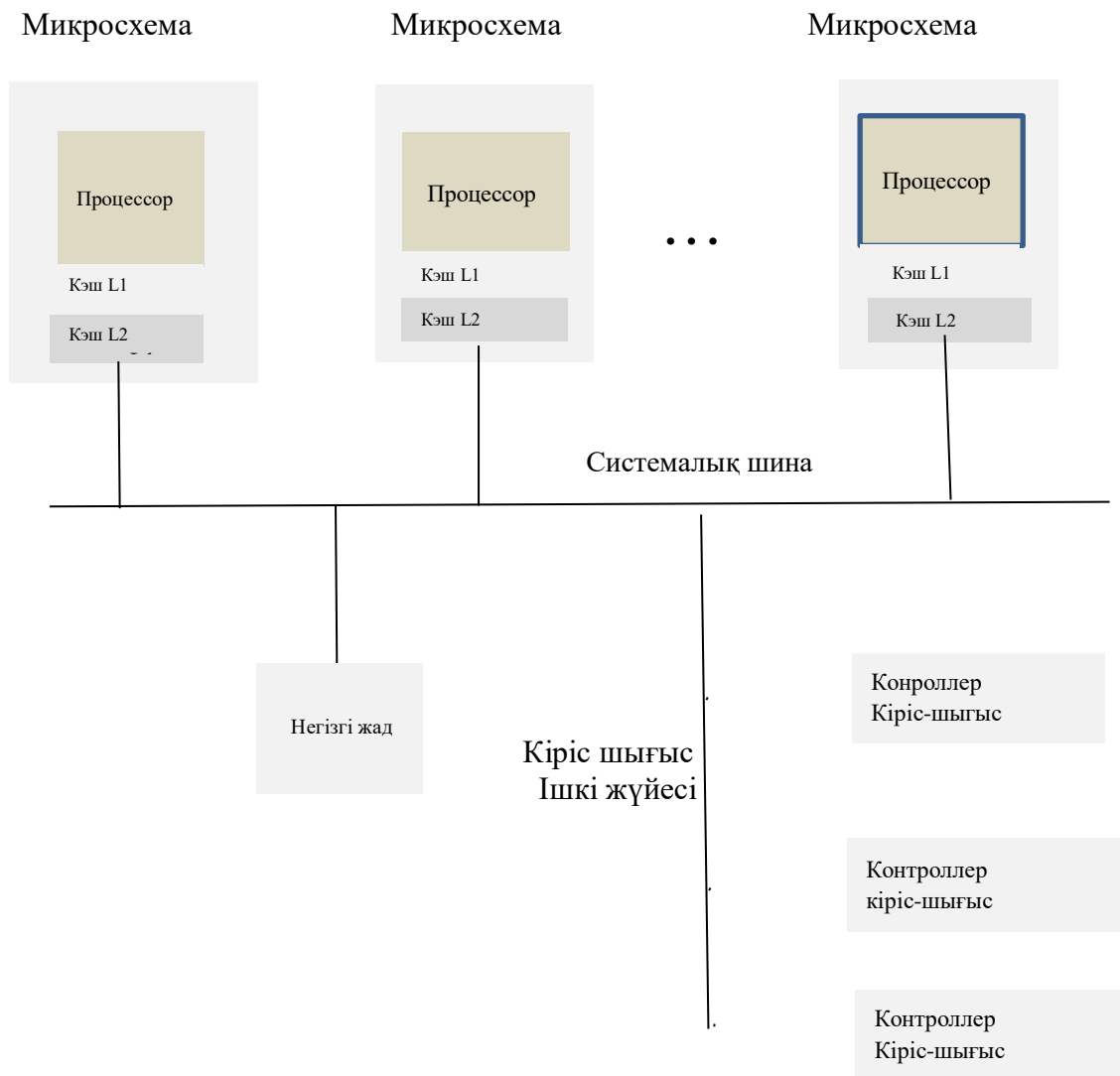
Сенімділік. Барлық процессорлар бірдей функцияларды орындай алатын симметриялы мультипроцессорлық жүйеде бір процессордың істен шығуы машинаны тоқтатпайды. Оның орнына жүйе төмен өнімділікпен жұмысын жалғастыра алады.

Өсу. Пайдаланушы қосымша процессорды қосу арқылы жүйенің жұмысын жақсарты алады.

Масштабтау. Жеткізушілер жүйеде орнатылған процессорлар санын өзгерту арқылы әртүрлі баға мен өнімділікке ие өнімдердің кең ауқымын ұсына алады.

Бірнеше процессорлардың болуы пайдаланушы үшін ашық. Операциялық жүйе жеке процессорлардың жұмысын жоспарлауды және процессорлар арасында синхрондауды өз мойнына алады.

Ұйымдастыру. 1.19-суретте SMP жалпы ұйымы көрсетілген. Бірнеше процессорлар бар, олардың әрқайсысында жеке басқару блогы, арифметикалық-логикалық құрылғы және регистрлер бар. Әдетте, әр процессорда L1 және L2 деп белгіленген екі арнайы кэш деңгейі болады.



1.9-сурет. –SMP ұйымдастыру

1.9-суретте көрсетілгендей, әр процессор және оның кэштері бөлек чипке орналастырылған. Әрбір процессор ортақ негізгі жадқа және кейбір байланыс механизмі арқылы енгізу-шығару құрылғыларына қол жеткізе алады; ортақ автобус ортақ объект болып табылады. Процессорлар бір-бірімен жад арқылы байланыса алады (хабарламалар мен күй туралы ақпарат ортақ мекен-жай кеңістігінде қалады). Процессорлардың сигналдарды тікелей бөлісуі де мүмкін. Жад көбінесе әртүрлі жад блоктарына бір уақытта бірнеше рет қол жеткізуге болатындай етіп ұйымдастырылады.

Қазіргі компьютерлерде процессорлар тек осы процессорға тиесілі кем дегенде бір кэш деңгейіне ие болады. Әрбір жергілікті кэш негізгі жадтың бір бөлігін көрсететіндіктен, бір кэштегі сөзді өзгерту басқа кэштегі сөзді жарамсыз етуі мүмкін. Бұл мәселе кэш сәйкестігі мәселесі ретінде белгілі және әдетте операциялық жүйе емес, аппараттық құралдар арқылы шешіледі.

ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Garg, R.; Verma, G. Operating Systems [OP]: An Introduction - Softcover Publisher: Mercury Learning & Information, 2017. 290 p.
2. <https://gifer.com/ru/7h0m>
3. <https://3dnews.ru/1034959>
4. Darrell Hajek, Cesar Herrera, Flor Narciso Principles of Operating Systems. Independently Published (24 April 2020) 176 pages.
5. Andrew S. Tanenbaum and Herbert Bos. Modern Operating Systems. 4/E. 1136 pages, Pearson India, 2016.
6. Silberschatz Abraham, Galvin Peter Baer and Gadne Greg. Operating system concepts.
7. Amdahl GM (1967) Validity of the single-processor approach to achieve large scale computing capabilities. AFIPS Joint Spring Conference Proceedings 30 (Atlantic City, NJ, Apr. 18–20), AFIPS Press, Reston VA, pp 483–485.
8. <https://studfile.net/>.
9. <https://habr.com/ru/post/40227/>.
10. wikimedia.org
11. wordpress.com
12. blackandwhitecomputer.blog
13. <http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf>.
14. encyclopedia2.thefreedictionary.com
15. linustechtips.com
16. youtube.com/watch?v=w3K1Jkly6D4