

Дәріс 2. Операциялық жүйелерге шолу

Тақырыптар:

- 2.1. Операциялық жүйелердің мақсаттары мен функциялары
- 2.2. Операциялық жүйелердің мақсаты
- 2.3. Заманауи операциялық жүйелерді жобалау
- 2.4. Ақауларға төзімділік
- 2.5. Операциялық жүйелерге шолу.
 - 2.5.1 Microsoft Windows
 - 2.5.2 Linux
 - 2.5.3 Android

2.1. ОПЕРАЦИЯЛЫҚ ЖҮЙЕЛЕРДІҢ МАҚСАТТАРЫ МЕН ФУНКЦИЯЛАРЫ

Операциялық жүйе-бұл қолданбалы бағдарламалардың орындалуын бақылайтын және қосымшалар мен компьютердің аппараттық құралдары арасындағы интерфейстің рөлін атқаратын бағдарлама. Оның мақсатын үш негізгі компонентке бөлуге болады.

* Ыңғайлылық. Операциялық жүйе компьютерді пайдалануды қарапайым және ыңғайлы етеді.

* Тиімділік. Операциялық жүйе компьютерлік жүйенің ресурстарын тиімді пайдалануға мүмкіндік береді.

* Даму мүмкіндігі. Операциялық жүйе жаңа қосымшалар мен жүйелік функцияларды тиімді әзірлеуге, сынауға және енгізуге мүмкіндік беретін етіп ұйымдастырылуы керек және бұл есептеу жүйесінің қалыпты жұмысына кедергі келтірмеуі керек.

2.1.1 Операциялық жүйе пайдаланушы мен компьютер арасындағы интерфейс ретінде

2.1-суретте соңғы пайдаланушыға қосымшалармен жұмыс істеу мүмкіндігін беру үшін қолданылатын бағдарламалық жасақтама мен аппараттық құралдардың құрылымы көрсетілген. Компьютер оны қосымшалар жиынтығы ретінде көреді. Жүйелік бағдарламалар жиынтығы бар, олардың кейбіреулері утилиталар немесе кітапхана бағдарламалары деп аталады. Олардың көмегімен пайдаланушы бағдарламаларын құруға, файлдармен жұмыс істеуге және енгізу-шығару құрылғыларын басқаруға көмектесетін жиі қолданылатын функциялар жүзеге асырылады. Қолданбалар жұмыс уақытында белгілі бір функцияларды орындау үшін утилиталарға жүгінеді. Жүйелік бағдарламалардың ішіндегі ең маңыздысы-бағдарламалық жасақтамадан аппараттық бөлшектерді жасыратын және оған жүйені пайдалану үшін ыңғайлы интерфейс беретін операциялық жүйе. Операциялық жүйе делдал ретінде әрекет етеді, бұл бағдарламашы мен бағдарламалық қосымшаларға әртүрлі қызметтер мен мүмкіндіктерге қол жеткізуді жеңілдетеді.



Сурет 2.1. Компьютердің бағдарламалық және техникалық құралдарының құрылымы (көшірілген)

Әдеттегі операциялық жүйелер ұсынатын қызметтердің қысқаша тізімі:

- Бағдарламаларды әзірлеу.
- Бағдарламаларды орындау.
- Енгізу-шығару құрылғыларына кіру.
- Файлдарға бақыланатын қол жетімділік.
- Жүйелік қол жетімділік.
- Қателерді анықтау және оларды өңдеу.
- Ресурстарды пайдалануды есепке алу.
- Командалар жүйесінің құрылымы (instruction set architecture - ISA).
- Қосымшаның екілік интерфейсі (application binary interface - ABI).
- Қолданбалы бағдарламалау интерфейсі (application programming interface - API).

2.1.2 Операциялық жүйе ресурстар менеджері ретінде

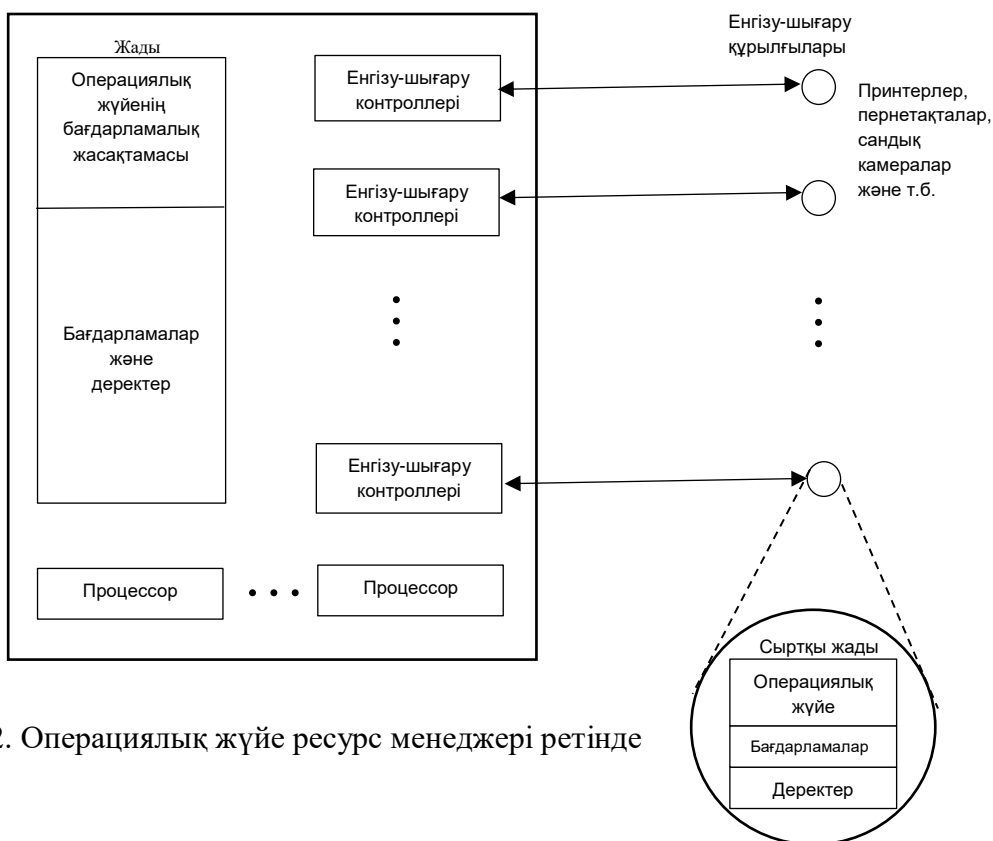
Операциялық жүйе енгізу-шығару, негізгі және қосымша жад және процессордың жұмыс уақыты сияқты компьютер ресурстарын пайдалануды басқаруға жауапты. Басқару механизмі ретінде операциялық жүйенің екі маңызды ерекшелігі бар:

Амалдық жүйенің функциялары барлық басқа бағдарламалық жасақтама сияқты жұмыс істейді, яғни олар жеке бағдарламалар немесе процессор орындайтын бағдарламалар жиынтығы түрінде жүзеге асырылады.

Амалдық жүйе көбінесе басқаруды басқа процестерге жібереді және процессордың өз міндеттерін қайтадан орындауға мүмкіндік беруін күту керек.

Кез-келген басқа бағдарлама сияқты, операциялық жүйе де процессор орындайтын командалардан тұрады. Жұмыс кезінде операциялық жүйе процессорға басқа жүйелік ресурстарды қалай пайдалану керектігін және басқа бағдарламаларды орындау кезінде уақытты қалай бөлу керектігін көрсетеді. 2.2-суретте операциялық жүйе басқаратын негізгі ресурстар көрсетілген. Амалдық жүйенің бір бөлігі негізгі жадта орналасқан. Бұл бөлікке ең көп қолданылатын функциялардың негізгі бөлігін қамтитын ядро кіреді; сол жерде қазіргі уақытта қолданылатын амалдық жүйенің кейбір басқа компоненттері де бар. Негізгі жадтың қалған бөлігінде басқа бағдарламалар мен пайдаланушы деректері бар.

Компьютерлік жүйе



Сурет 2.2. Операциялық жүйе ресурс менеджері ретінде

2.1.3 Операциялық жүйені дамыту

Көптеген операциялық жүйелер үнемі дамып отырады. Бұл келесі себептерге байланысты болады.

- * Аппараттық құралдардың жаңа түрлерін жаңарту және пайда болу.
- * Жаңа қызметтер.
- * Түзетулер.

Операциялық жүйелерді үнемі өзгерту қажеттілігі олардың құрылғысына белгілі бір талаптар қояды. Әлбетте, бұл жүйелер модульдердің өзара әрекеттесуі жақсы анықталған модульдік дизайнға ие болуы керек; жақсы және толық құжаттама өте маңызды рөл атқарады.

2.2. Операциялық жүйелердің мақсаты

Операциялық жүйелер ең күрделі бағдарламалардың қатарына жатады. Бұл олардың әзірлеушілерінің жүйелерді ыңғайлылық пен тиімділік талаптарын қанағаттандыратын және әлі де даму қабілетін жоғалтпайтын етіп жасауға деген ұмтылысын көрсетеді. [64] сәйкес операциялық жүйелерді дамыту процесінде төрт негізгі бағытта зерттеулер жүргізілді.

- * Процестер
- * Жадты басқару
- * Ақпаратты қорғау және қауіпсіздік
- * Ресурстарды жоспарлау және басқару

Осы бағыттардың әрқайсысын күрделі практикалық мәселелерді шешуге арналған дерексіз принциптер жиынтығымен сипаттауға болады. Негізінен, қазіргі заманғы операциялық жүйелердің дамуы жоғарыда аталған бағыттар бойынша жүреді.

Процестерді басқару. Операциялық жүйелерді жобалаудың негізгі тұжырымдамаларының бірі-процестер тұжырымдамасы. Бұл терминді алғаш рет 1960 жылдары multics операциялық жүйесін жасаушылар қолданған [56]. Процесс терминінің көптеген анықтамалары бар, соның ішінде:

- * орындалатын бағдарлама;
- * компьютерде орындалатын бағдарламаның данасы;
- * процессорда анықтауға және орындауға болатын нысан;
- * бірізді әрекеттер тізбегімен, ағымдағы күймен және онымен байланысты жүйелік ресурстар жиынтығымен сипатталатын белсенділік бірлігі.

Жадты басқару. Пайдаланушының қажеттіліктерін модульдік бағдарламалауды және деректерді икемді пайдалануды қолдайтын есептеу ортасы жақсы қанағаттандырады. Операциялық жүйенің басқару бағдарламалары тарапынан деректерді сақтау құрылғысына орналастыруды тиімді және жүйелі бақылауды қамтамасыз ету қажет. Тұжырымдалған талаптарға сүйене отырып, операциялық жүйе осындай функцияларды орындауы керек.

1. Процестерді оқшаулау. Операциялық жүйе тәуелсіз процестердің ешқайсысы басқа процеске бөлінген жад мазмұнын өзгерте алмайтындығына көз жеткізуі керек және керісінше.

2. Автоматты орналастыру және басқару. Бағдарламалар белгілі бір талаптарға сәйкес жадта динамикалық түрде орналастырылуы керек. Жадты бөлу бағдарламашы үшін ашық болуы керек. Осылайша, бағдарламашы жад аясына байланысты шектеулерді қадағалап отыру қажеттілігінен арылады, ал Операциялық жүйе тапсырмаларға тек қажетті жад көлемін бөлу арқылы есептеу жүйесінің тиімділігін арттырады.

3. Модульдік бағдарламалауды қолдау. Бағдарламашы бағдарламаның модульдерін анықтай алуы керек, сонымен қатар оларды динамикалық түрде құрып, жойып, өлшемін өзгерте алады.

4. Кіруді қорғау және басқару. Жадты әр иерархиялық деңгейде бөлісу кезінде бір бағдарламаның басқа бағдарламаның жад кеңістігіне жүгіну мүмкіндігі бар. Мұндай мүмкіндік, егер ол осы қосымшаның жұмыс принципіне енгізілген болса, қажет болуы

мүмкін. Екінші жағынан, бұл бағдарламалардың тұтастығына және операциялық жүйенің өзіне қауіп төндіреді. Операциялық жүйе әртүрлі пайдаланушылардың жадтың әртүрлі аймақтарына қалай қол жеткізе алатынын бақылауы керек.

5. Ұзақ мерзімді сақтау. Көптеген қосымшалар компьютерді өшіргеннен кейін ақпаратты ұзақ уақыт сақтауға болатын құралдарды қажет етеді.

Әдетте, операциялық жүйелер бұл талаптарды виртуалды жад құралдары мен файлдық жүйе арқылы орындайды. Файлдық жүйе файлдар деп аталатын нысандарға орналастырылған ақпараттың ұзақ мерзімді сақталуын қамтамасыз етеді. Файл-бұл бағдарламашыға ыңғайлы тұжырымдама, оған қол жеткізу және қорғау амалдық жүйемен жүзеге асырылады. Виртуалды жад - бұл бағдарламашыларға жадты логикалық тұрғыдан қарастыруға мүмкіндік беретін функционалды мүмкіндік, физикалық жадтың жеткілікті көлемінің болуына мән бермейді.

Ақпаратты қорғау және қауіпсіздік. Қауіпсіздік пен ақпаратты қорғау міндеттерінің көпшілігін шартты түрде төрт санатқа бөлуге болады.

1. Қол жеткізу. Жүйені сыртқы араласудан қорғаумен байланысты.
2. Құпиялылық. Рұқсат етілмеген пайдаланушының деректерді оқи алмауына кепілдік береді.
3. Деректердің тұтастығы. Деректерді рұқсат етілмеген өзгерістерден қорғау.
4. Аутентификация. Пайдаланушылардың тиісті тексерілуін және хабарламалардың немесе деректердің дұрыстығын қамтамасыз ету.

Ресурстарды жоспарлау және басқару. Операциялық жүйенің маңызды міндеттерінің бірі-оның қолындағы ресурстарды басқару (негізгі жад, енгізу-шығару құрылғылары, процессор), сондай-ақ оларды әртүрлі белсенді процестер арасында пайдалануды жоспарлау. Ресурстарды бөлу стратегиясын әзірлеу кезінде келесі факторларды ескеру қажет.

1. Теңдік. Әдетте, белгілі бір ресурсты талап ететін барлық процестерге оған бірдей қол жетімділік берілгені жөн.
2. Жауаптың дифференциациясы. Операциялық жүйе талаптардың бүкіл жиынтығына сәйкес ресурстарды бөлуді жүзеге асырды.

Тиімділік. Операциялық жүйе жүйенің өткізу қабілеттілігін арттыруы керек, оның жауап беру уақытын азайтуы керек және егер ол уақытты бөлу жүйесінде жұмыс істесе, пайдаланушылардың ең көп санына қызмет етуі керек.

2.3. Заманауи операциялық жүйелерді жобалау

- Операциялық жүйелерге қойылатын талаптардың өсуі олардың архитектурасын жетілдіруге ғана емес, сонымен қатар оларды ұйымдастырудың жаңа әдістерінің пайда болуына әкеледі.

Қазіргі операциялық жүйелерде зерттелген тәсілдер мен құрылымдық элементтерді келесі санаттарға біріктіруге болады:

- Микро ядро архитектурасы.
- Көп ағынды.
- Симметриялы мультипроцессорлық.
- Таратылған операциялық жүйелер.
- Нысанға бағытталған дизайн.
- Көптеген операциялық жүйелердің айрықша ерекшелігі-үлкен монолитті ядро. Операциялық жүйенің ядросы оның көптеген мүмкіндіктерін ұсынады, соның ішінде жоспарлау, файлдық жүйемен жұмыс істеу, желі функциялары, әртүрлі құрылғы драйверлерінің жұмысы, жадты басқару және т.б. Микро ядро архитектурасында ядроға мекен - жай кеңістігін басқару, процестер арасындағы өзара әрекеттесуді қамтамасыз ету (interprocess communication-IPC) және негізгі жоспарлауды қамтитын бірнеше маңызды функциялар ғана беріледі. Операциялық жүйенің басқа қызметтері кейде серверлер деп аталатын процестермен қамтамасыз етіледі. Бұл процестер пайдаланушы режимінде жұмыс істейді және микро ядро

олармен басқа қолданбалар сияқты жұмыс істейді.

- Көп ағынды (multithreading) - бұл қосымшаны орындайтын процесс бір уақытта бірнеше ағынға бөлінетін технология. Төменде ағын мен процесс арасындағы негізгі айырмашылықтар берілген.
- Ағын (жіп). Процессордың контекстін (командалық есептегіштің және стек шыңының көрсеткішінің мазмұнын қамтитын), сондай-ақ өзінің стек аймағын (ішкі бағдарламаларды шақыруды ұйымдастыру үшін) қамтитын диспетчерлік жұмыс бірлігі. Ағын командалары дәйекті түрде орындалады; процессор басқа ағынды өңдеуге ауысқан кезде ағын үзілуі мүмкін.
- Процесс. Бір немесе бірнеше ағындардың жиынтығы, сондай-ақ осы ағындармен байланысты жүйелік ресурстар (мысалы, код пен деректерді, ашық файлдарды, әртүрлі құрылғыларды қамтитын жад аймағы). Бұл тұжырымдама орындалатын бағдарлама тұжырымдамасына өте жақын. Қолданбаны бірнеше ағынға бөлу арқылы бағдарламашы қолданбаның модульділігінің барлық артықшылықтарын және қолданбаға қатысты уақытша оқиғаларды басқару мүмкіндігін алады.

Көп ағынды дәйекті орындауды қажет етпейтін бірнеше тәуелсіз тапсырмаларды орындайтын қосымшалар үшін өте пайдалы. Көп ағынды және мультипроцессорлық ұғымдар тәуелсіз. Көп ағынды-бұл бір процессорлы машинада да қосымшалар мен ядро процестерін құрылымдаудың пайдалы тұжырымдамасы. Екінші жағынан, процестер бірнеше ағындарға бөлінбесе де, мультипроцессорлық жүйе бір процессорға қарағанда артықшылықтарға ие болуы мүмкін, өйткені мұндай жүйеде бір уақытта бірнеше процестерді іске қосуға болады. Көп ағынды және көп процессорды бөлісу айтарлықтай әсер етеді. Мультипроцессорлық жүйелердің ерекшелігі-бірнеше процессорлардың болуы пайдаланушы үшін ашық - барлық процестер мен процессорлар пайдаланушыға бірыңғай жүйе түрінде көрінді, ал операциялық жүйе ағындардың процессорлар арасында бөлінуіне және әртүрлі процестердің синхрондалуына жауап береді.

Жоғары деңгейдегі міндет - бірнеше бөлек компьютерлердің бір кластерлік жүйесі ретінде ұсыну. Бұл жағдайда біз әрқайсысының өзіндік негізгі және қосалқы жады және өзінің енгізу-шығару модульдері бар компьютерлер жиынтығымен айналысамыз. Таратылған операциялық жүйе негізгі және қосалқы жадтың, сондай-ақ бірыңғай файлдық жүйенің бір кеңістігінің елесін жасайды.

Операциялық жүйелерді орнатуға соңғы жаңалықтардың бірі объектіге бағытталған технологияларды қолдану болды. Операциялық жүйе деңгейінде объектілерге негізделген құрылым бағдарламашыларға операциялық жүйені оның тұтастығын бұзбай конфигурациялауға мүмкіндік береді. Сонымен қатар, бұл тәсіл таратылған құралдар мен толыққанды таратылған операциялық жүйелердің дамуын жеңілдетеді.

2.4 Ақауларға төзімділік

Ақауға төзімділік дегеніміз-аппараттық немесе бағдарламалық жасақтама қателіктеріне қарамастан жүйенің немесе компоненттің қалыпты жұмысын жалғастыру мүмкіндігі. Ақауларға төзімділік жүйенің сенімділік дәрежесін арттыруға арналған. Ақауларға төзімділіктің артуы қаржылық немесе өнімділіктің төмендеуімен көрсетілген белгілі бір мәнге ие.

Жүйенің жұмыс істеу сапасының ақаулыққа төзімділікпен байланысты үш негізгі көрсеткіші бар: сенімділік, істен шығудың орташа уақыты және қол жетімділік. Бұл тұжырымдамалар аппараттық ақауларға ерекше назар аударатын отырып жасалған, бірақ тұтастай алғанда аппараттық және бағдарламалық жасақтаманың бұзылуына қолданылады.

IEEE стандарттарының сөздігі сәтсіздікті (fault) кейбір компоненттің істен шығуы, оператор қатесі, қоршаған ортаның физикалық кедергісі, дизайн қатесі, бағдарламалау немесе деректер құрылымы нәтижесінде аппараттық немесе бағдарламалық жасақтаманың қате күйі ретінде анықтайды. Стандарт сонымен қатар сәтсіздік 1) аппараттық құрылғының немесе компоненттің ақауы немесе 2) компьютерлік бағдарламадағы дұрыс

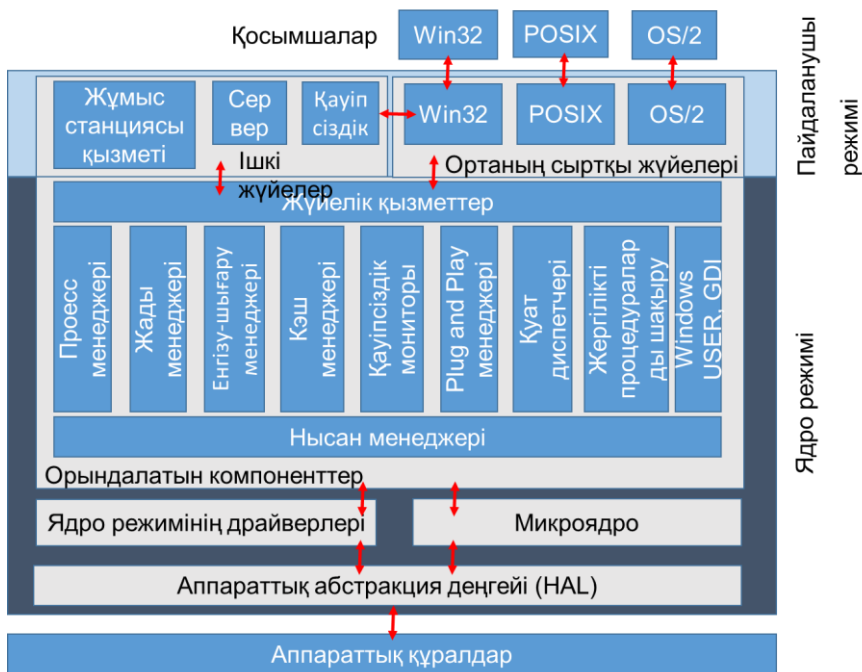
емес әрекет, процесс немесе деректер ретінде көрінетінін анықтайды.

Ақауларға төзімділікті қолдаудың бірқатар әдістері операциялық жүйенің бағдарламалық жасақтамасына енгізілуі мүмкін. Ақаулыққа төзімділікті жүзеге асырудың мысалдары процестерді оқшаулау болып табылады: процестерді басқару үшін операциялық жүйе ұсынатын құрылым басқа процестердің сәтсіз процесінен қорғаудың белгілі бір деңгейін қамтамасыз етеді; параллелизмді басқару: дұрыс жұмыс істеуді қамтамасыз ету және сәтсіздіктерден кейін қалпына келтіру үшін қолданылатын әдістер, мысалы, өзара блоктау; қалпына келтіру нүктелері және кері қайтару: апат болған кезде қолданба күйі алдыңғы қалпына келтіру нүктесіне оралып, қайта іске қосылады.

2.5. Операциялық жүйелерге шолу.

2.5.1 Microsoft Windows

2.3-суретте Windows-тың жалпы құрылымы көрсетілген. Windows қолданбалы бағдарламалық жасақтаманы амалдық жүйенің өзегінен бөледі. Соңғысына ядро режимінде жұмыс істейтін орындау жүйесі, микро ядро, құрылғы драйверлері және аппараттық абстракция деңгейі (hardware abstraction layer - HAL) кіреді. Осы режимде орындалатын бағдарламалар жүйелік деректерге және аппараттық құралдарға қол жеткізе алады. Пайдаланушы режимінде жұмыс істейтін басқа бағдарламалардың жүйелік деректерге қол жетімділігі шектеулі.



Сурет 2.3. Windows архитектурасы [212](көшірілген)

Windows модульдерге нақты бөлінуге тән. Әрбір жүйе функциясы тек бір операциялық жүйенің құрамдас бөлігімен басқарылады. Оның қалған бөліктері мен барлық қосымшалары бұл функцияға стандартты интерфейс арқылы қол жеткізеді. Негізгі жүйелік деректерге тек белгілі бір функциялар арқылы қол жеткізуге болады.

Төменде ядро режимінде жұмыс істейтін Windows компоненттері берілген:

Атқарушы жүйе. Жадты, процестерді және ағындарды басқару, қауіпсіздік, енгізу-шығару және процессаралық өзара әрекеттесу сияқты негізгі операциялық жүйе қызметтерін қамтиды.

Ядро. Процессорлардың жұмысын басқарады. Ядро ағындарды жоспарлауды, процестерді ауыстыруды, ерекшеліктер мен үзілістерді өңдеуді және көп процессорлы синхрондауды басқарады.

Аппараттық абстракция деңгейі (hardware abstraction layer – HAL). Жалпыланған

командалар мен аппараттық жауаптарды бірегей командаларға және белгілі бір платформаның аппараттық жауаптарына көрсетеді. Операциялық жүйені аппараттық платформалардың айырмашылықтарынан оқшаулайды. HAL жүйелік шинаны, тікелей қол жетімді жад контроллерін (DMA), үзіліс контроллерін, жүйелік таймерлерді және әр түрлі компьютерлердің жад контроллерін атқарушы жүйе мен ядро компоненттеріне ұқсас етеді.

Құрылғы драйверлері. Атқарушы жүйенің функционалдығын кеңейтетін динамикалық кітапханалар. Оларға белгілі бір аппараттық құрылғыларға сұраныстарға енгізу-шығару функцияларының пайдаланушы қоңырауларын жіберетін аппараттық құрылғы драйверлері және файлдық жүйелерді, желілік протоколдарды және ядро режимінде орындалуы керек басқа жүйелік кеңейтімдерді іске асырудың бағдарламалық компоненттері кіреді.

Терезе және графикалық жүйелер. Терезелермен жұмыс істеу, пайдаланушы интерфейсін басқару және экранға шығару сияқты графикалық пайдаланушы интерфейсінің функцияларын жүзеге асырады.

Windows пайдаланушы режимінің процестерінің төрт негізгі түрін қолдайды.

1. Арнайы жүйелік процестер. Мұндай процестерге Windows амалдық жүйесіне кірмеген утилиталар кіреді, мысалы, кіру процесі, аутентификация жүйесі және сессия менеджері.

2. Қызметтік процестер. Принтер тапсырмаларының кезегі, оқиғаларды жазу, құрылғы драйверлерімен өзара әрекеттесуге арналған теңшелетін компоненттер, әртүрлі желілік қызметтер және т.б.

3. Қоршаған ортаның ішкі жүйелері. Windows қызметтерін пайдаланушының қосымшаларын қамтамасыз етіңіз, осылайша амалдық жүйенің ортасын қамтамасыз етіңіз. Win32 және POSIX сияқты ішкі жүйелерге қолдау көрсетіледі. Әрбір ортаның ішкі жүйесіне динамикалық қосылатын кітапханалар кіреді, олар пайдаланушының қолданбалы қоңырауларын ALPC қоңырауларына түрлендіреді (кеңейтілген жергілікті процедуралық қоңырау - advanced local procedure call, ALPC) және/немесе Windows қоңыраулары.

4. Арнайы қолданбалар. Орындалатын файлдар (eche) және динамикалық қосылатын кітапханалар (DLL) пайдаланушыларға жүйені қолдану мүмкіндігін береді.

Ағындарды қолдау және симметриялы мультипроцессорлық қолдау мүмкіндіктері Windows амалдық жүйесінің маңызды сипаттамалары болып табылады. Төменде Windows амалдық жүйесіндегі негізгі ағындық қолдау және SMP мүмкіндіктері келтірілген [212]:

* Операциялық жүйенің утилиталары кез-келген бос процессорларда жұмыс істей алады; әр түрлі бағдарламалар бір уақытта әр түрлі процессорларда жұмыс істей алады.

* Windows амалдық жүйесі бірнеше ағындарға бөлінген бір процесті орындауды қолдайды. Бұл ағындарды бір уақытта бірнеше процессорларда орындауға болады.

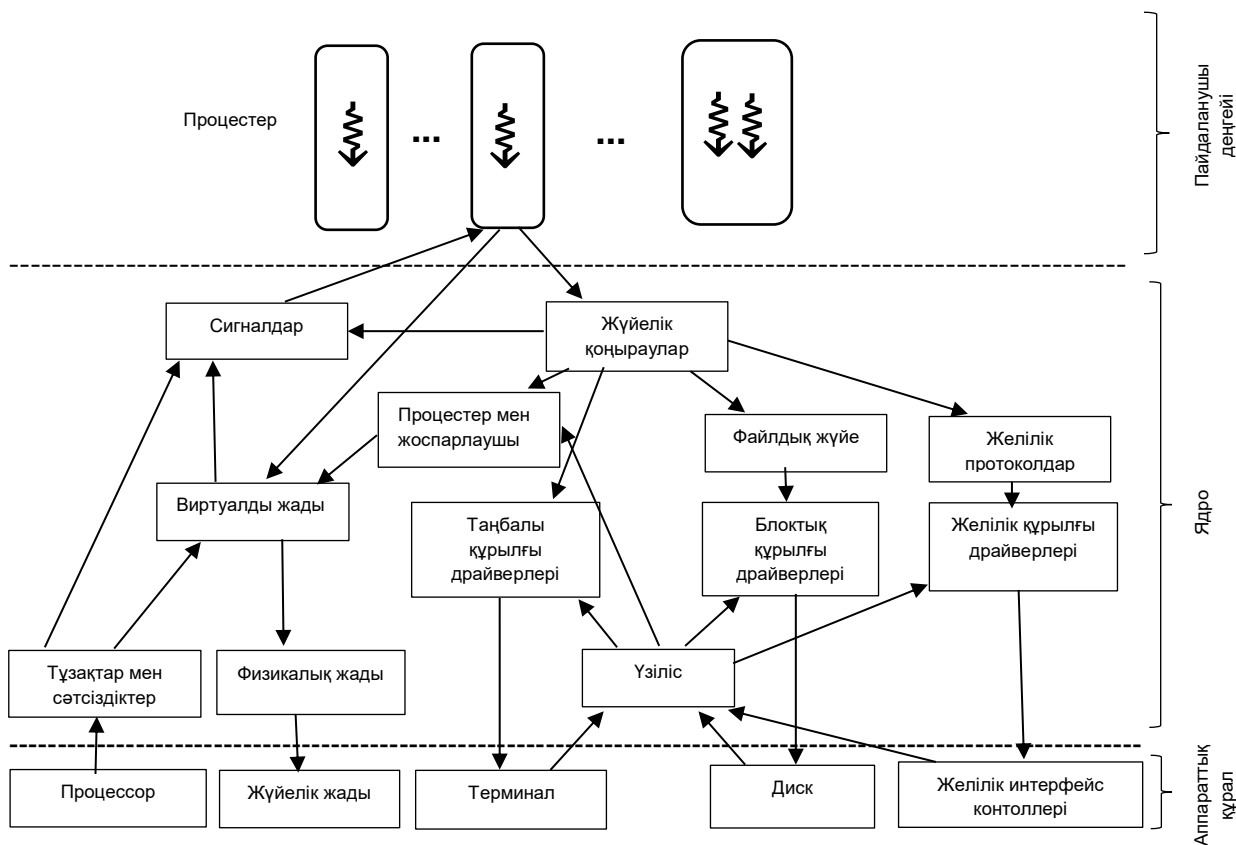
* Серверлік процестер әр түрлі клиенттерден келетін сұраныстарды бір уақытта өңдеу үшін бірнеше ағындарды қолдана алады.

* Windows операциялық жүйесі әртүрлі процестермен деректер мен ресурстарды бөлісу механизмдерін, сондай-ақ процестер арасында ақпарат алмасудың икемді мүмкіндіктерін ұсынады.

2.5.2 Linux

Linux микро ядро әдісін қолданбайды, ол өзінің модульдік архитектурасы арқылы осы тәсілдің көптеген әлеуетті артықшылықтарына қол жеткізеді. Linux операциялық жүйесі модульдер жиынтығы ретінде ұйымдастырылған, олардың қатары сұраныс бойынша автоматты түрде жүктелуі мүмкін. Бұл салыстырмалы түрде тәуелсіз блоктар жүктелетін модульдер деп аталады [92]. 2.4-суретте әдеттегі Linux ядросының негізгі компоненттері көрсетілген. Суретте ядроның үстінде жұмыс істейтін бірнеше процестер көрсетілген. Әрбір жақтау жеке процесті көрсетеді, ал жақтаудағы белгі бар әрбір толқынды сызық орындалу ағынын білдіреді. Ядроның өзі осы компоненттердің негізгі

өзара әрекеттесуін көрсететін белгілері бар өзара әрекеттесетін компоненттер жиынтығынан тұрады. Негізгі аппараттық құралдар сонымен қатар ядроның қандай компоненттері қандай аппараттық компоненттерді қолданатынын немесе басқаратынын көрсететін белгілері бар компоненттер жиынтығы ретінде бейнеленген.



Сурет 2.4. Linux ядросының компоненттері

Ядроның негізгі компоненттері:

- Сигналдар. Ядро процеске қол жеткізу үшін сигналдарды пайдаланады.
- Жүйелік қоңыраулар. Жүйелік қоңырау-бұл процесс белгілі бір ядро қызметін сұрайтын құрал.
- Процестер және жоспарлаушы. Процестерді жасайды, басқарады және жоспарлайды.
- Виртуалды жад. Процестер үшін виртуалды жадты бөледі және басқарады.
- Файлдық жүйе. Файлдарға, каталогтарға және файлдық жүйенің файлдары мен функцияларына қатысты басқа нысандарға ғаламдық иерархиялық атауар кеңістігін ұсынады.
- Желілік хаттамалар.
- Таңбалық құрылғы драйверлерін бағыттау. Терминалдар, принтерлер немесе модемдер сияқты деректерді бір байт бойынша жіберуді немесе алуды қажет ететін құрылғыларды басқару.
- Блоктық құрылғы драйверлері. Деректерді блоктарда оқитын және жазатын құрылғыларды басқару, мысалы, қайталама жадтың әртүрлі түрлері.
- Желілік құрылғы драйверлері. Көпірлер немесе маршрутизаторлар сияқты желілік құрылғыларға қосылатын желілік интерфейс карталарын және байланыс

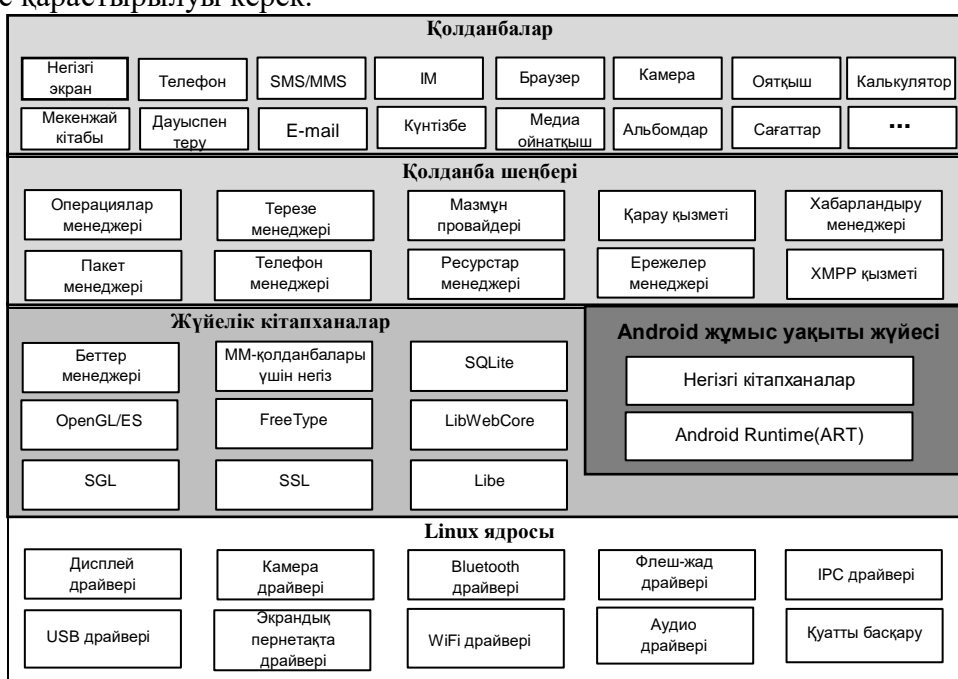
порттарын басқару.

- Тұзақтар мен сәтсіздіктер. Процессор жасаған үзілістерді өңдеу, мысалы, жад істен шыққан кезде.
- Физикалық жады. Бет кадрларының пулын басқарады және виртуалды жадқа беттерді бөледі.
- Үзіліс. Перифериялық құрылғылардан үзілістерді өңдеу.

2.5.3 Android

Android операциялық жүйесі Linux операциялық жүйесіне негізделген және бастапқыда ұялы телефондарға арналған. Бұл операциялық жүйе серверлер мен дербес компьютерлерді қоспағанда, компьютерлік чипі бар кез-келген құрылғыда жиі орнатылады. Android-интернет заттары үшін кеңінен қолданылатын операциялық жүйе.

Android-бұл Linux ядросының өзгертілген нұсқасын, аралық бағдарламалық жасақтаманы және негізгі қосымшаларды қамтитын бағдарламалар стегі. 2.6-суретте Android бағдарламалық жасақтамасының архитектурасы толығырақ көрсетілген. Осылайша, Android операциялық жүйе ретінде ғана емес, толық бағдарламалық жасақтама ретінде қарастырылуы керек.



Жүзеге асыру:

Қолданбалар, қолданбалы жүйе: Java

Жүйелік кітапханалар, Android жұмыс уақыты жүйесі: C, C++

Linux ядросы: C

Сурет 2.5. Android бағдарламалық архитектурасы

Пайдаланушы тікелей өзара әрекеттесетін барлық қолданбалар қолданба деңгейінің бөлігі болып табылады. Бұған электрондық пошта клиенті сияқты жалпы мақсаттағы қосымшалардың негізгі жиынтығы кіреді, SMS-бағдарлама, күнтізбе, геолокациялық карталар, шолғыш, контактілер кітабы және кез-келген мобильді құрылғыға стандартты басқа қосымшалар. Қолданбалар әдетте Java тілінде жүзеге асырылады.

Қолданбалы жақтау деңгейі (Application Framework) бағдарламашы жаңа қолданбаларды жасау үшін пайдалана алатын стандартталған API арқылы жоғары деңгейлі "құрылыс блоктарын" қамтамасыз етеді. Архитектура компоненттерді қайта пайдалануды мүмкіндігінше жеңілдетуге арналған. Application framework астындағы деңгей екі бөліктен тұрады: жүйелік кітапханалар және Android жұмыс уақыты жүйесі.

Жүйелік кітапхана-бұл жазылған пайдалы жүйелік функциялар жиынтығы C немесе C++ және әр түрлі Android компоненттерінде қолданылады.

Android операциялық жүйесінің ядросы стандартты Linux дистрибутивінің ядросы негізінде жасалған. Android қауіпсіздік, жадты басқару, процесті басқару, желілік стек және драйвер моделі сияқты негізгі жүйелік қызметтер жағдайында Linux ядросына негізделген. Android ядросында мобильді ортада қажет емес драйверлер жоқ, бұл ядроны кішірек етеді. Android Linux ядросын мобильді ортада жұмыс істеуге арналған мүмкіндіктермен кеңейтеді.

IOS және Windows сияқты мобильді құрылғыларда қолданылатын операциялық жүйелердің көпшілігі белгілі бір аппараттық платформа үшін екілік кодқа тікелей құрастырылатын бағдарламалық жасақтаманы пайдаланады. Бұл тәсілден айырмашылығы, Android бағдарламалық жасақтамасының көп бөлігі байткодта көрсетіледі, содан кейін ол құрылғының өзінде машиналық командаларға айналады.

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](https://www.wikimedia.org/)
11. [wordpress.com](https://www.wordpress.com/)
12. blackandwhitecomputer.blog
13. <http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf>.
14. encyclopedia2.thefreedictionary.com
15. [linustechtips.com](https://www.linustechtips.com/)
16. [youtube.com/watch?v=w3K1JkIY6D4](https://www.youtube.com/watch?v=w3K1JkIY6D4)