Дәріс 5. Ағындар

5.1 Процестер мен ағындар
5.2 Ағындардың түрлері
5.3 Көп ядролы және көп ағынды
5.4 Windows жүйесінде процестер мен ағындарды басқару
5.5 Android-де процестер мен ағындарды басқару

5.1 Процестер мен ағындар

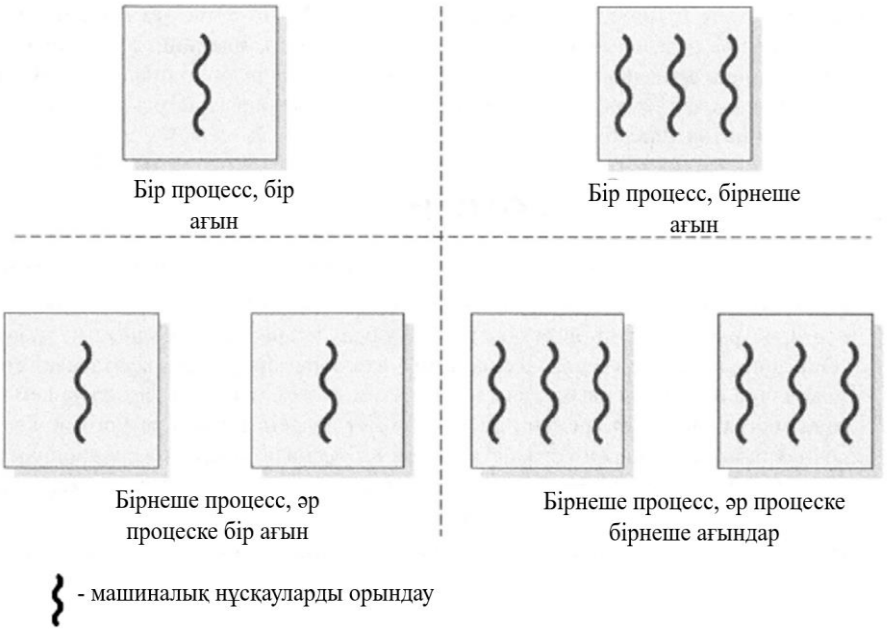
Процесс тұжырымдамасы екі бөлек, ықтимал тәуелсіз тұжырымдамаларды біріктіреді, олардың біреуі ресурстарды иеленумен, ал екіншісі процестерді орындаумен байланысты. Көптеген операциялық жүйелерде бұл айырмашылық ағын (thread) деп аталатын құрылымның пайда болуына әкеледі.

Осы уақытқа дейін процесс тұжырымдамасын екі қасиетпен сипаттауға болады:

- Ресурстарға иелік ету (resource ownership).

- Жоспарлау / орындау (жоспарлау/орындау). Процесс бір немесе бірнеше бағдарламалардың кодын орындау арқылы жүзеге асырылады. Бұл процестің орындалуы басқа процестердің орындалуымен ауысуы мүмкін. Сондықтан процесс диспетчерлеудің күйі мен басымдығы сияқты параметрлерге ие және операциялық жүйе жоспарлау мен диспетчерлеуді жүзеге асыратын субъект болып табылады.

Ағындар әдетте ағын (thread) немесе жеңіл процесс (lightweight process) деп аталады, ал ресурстарды иелену бірлігі процесс (process) немесе тапсырма (task) деп аталады. Көп ағынды (multithreading) - бұл операциялық жүйенің бір процесте бірнеше параллель орындау жолдарын қолдау мүмкіндігі. Әрбір процесс орындаудың бір ағыны болатын дәстүрлі тәсіл- бір ағынды тәсіл деп аталады. 5.1-суреттің екі сол жақ бөлігінде бір ағынды тәсілдер көрсетілген. Бір ағынды пайдаланушы процесін қолдауға қабілетті операциялық жүйенің мысалы MS-DOS болып табылады. Unix-тің әртүрлі сорттары сияқты басқа операциялық жүйелер көптеген пайдаланушылардың процестерін қолдайды, бірақ олардың әрқайсысында тек бір ағын болуы мүмкін. Суреттің оң жақ жартысында. 5.1 көп ағынды тәсілдерді ұсынылған. Бір процесс бірнеше ағындарға бөлінетін жүйенің мысалы, Java орындалу ортасы болып табылады. Көп ағынды ортада процесс ресурстарды бөлудің құрылымдық бірлігі, сондай-ақ қорғаудың құрылымдық бірлігі ретінде анықталады.



Сурет 5.1. Ағындар мен процестер

Келесі элементтер процестермен байланысты:

- Процестің кескіні бар виртуалды мекен-жай кеңістігі.

- Процессорларға, басқа процестерге, файлдарға және енгізу-шығару ресурстарына қауіпсіз қол жеткізу.

Процесс аясында бір немесе бірнеше ағындар болуы мүмкін, олардың әрқайсысы келесі сипаттамаларға ие.

- Ағынның орындалу жағдайы (орындалатын, орындауға дайын және т.б.).

- Орындалмаған ағынның сақталған контексті; ағынды қарастырудың бір әдісі-оны процесс аясында жұмыс істейтін тәуелсіз командалық санауыш ретінде қарастыру.

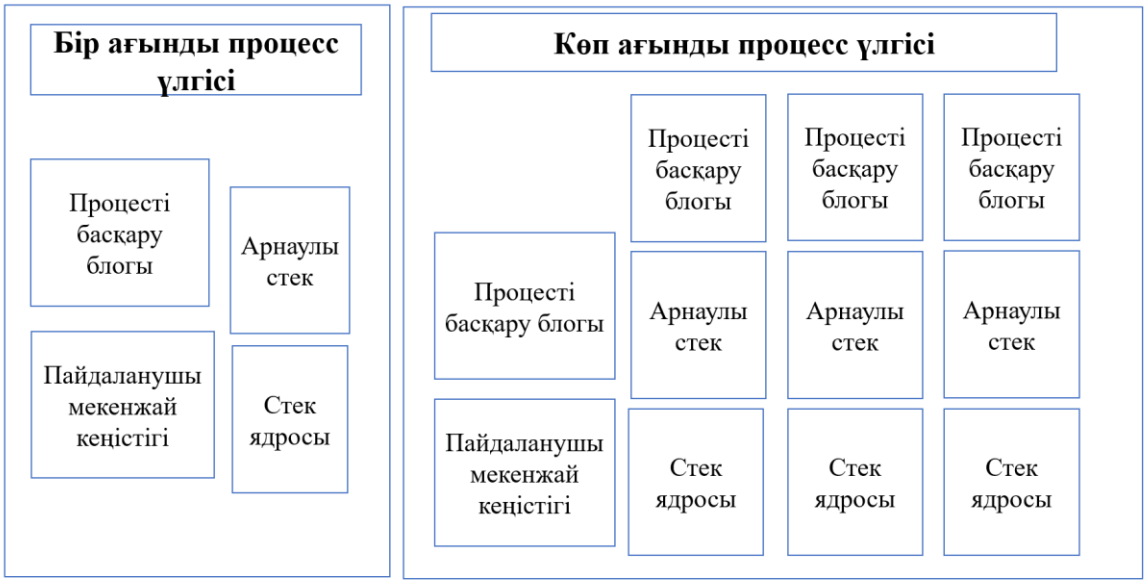
- Орындау стегі.

- Жергілікті айнымалылар үшін ағынға бөлінетін статикалық жад.

- Осы ағынға тиесілі жад пен технологиялық ресурстарға қол жеткізу; бұл қатынас осы процестің барлық ағындарымен бөлінеді.

5.2-суретте ағындар мен процестердің басқарылуы тұрғысынан айырмашылығы көрсетілген. Процестің бір ағынды үлгісінде оның көрсетілімі процестің басқару блогы мен пайдаланушының мекенжай кеңістігін, сондай-ақ процесс орындалып жатқанда процедураларды шақыру және одан қайтару үшін пайдаланылатын ядро мен пайдаланушы стектерін қамтиды. Процесс жұмыс істеп тұрған кезде ол процессор регистрлерін басқарады. Процесс үзілген кезде процессор регистрлерінің мазмұны жадта сақталады. Көп ағынды ортада жалғыз басқару блогы мен мекен-жай кеңістігі әр процеспен байланысты, бірақ қазір әр ағын үшін жеке стектер, сондай-ақ регистр мәндері, басымдық және ағынның күйі туралы басқа ақпарат бар басқару блогы

жасалады. Осылайша, процестегі барлық ағындар сол процестің күйі мен ресурстарын бөліседі. Олар бір мекенжай кеңістігінде және бірдей деректерге қол жеткізе алады. Егер бір ағын жадтағы кейбір деректерді өзгертсе, онда басқа ағындардың осы деректерге қатынасу кезінде осы өзгерістерді қадағалау мүмкіндігі болады. Бір ағын оқу рұқсаты бар файлды ашса, процестегі басқа ағындар да сол файлдан оқи алады.



Сурет 5.2. Бір ағынды және көп ағынды процесс модельдері

Өнімділік тұрғысынан ағындарды қолданудың негізгі артықшылықтары келесідей.

1. Қолданыстағы процесте жаңа ағынды құру үшін жаңа процесті құруға қарағанда аз уақыт қажет.

2. Ағынды процесске қарағанда тезірек аяқтауға болады.

3. Бір процесс аясында ағындар арасында ауысу процестер арасында ауысуға қарағанда әлдеқайда жылдам.

4. Ағындарды пайдалану кезінде орындалатын екі бағдарлама арасында ақпарат алмасудың тиімділігі артады.

Ағындарды сәтті қолдануға болатын қосымшаның мысалы-файлдық сервер. Әрбір жаңа файл сұранысын алған кезде файлдарды басқару бағдарламасы жаңа ағынды тудыруы мүмкін. Сервер өте көп сұраныстарды өңдеуі керек болғандықтан, қысқа уақыт ішінде көптеген ағындар жасалады және жойылады. Егер мұндай серверлік бағдарлама мультипроцессорлық машинада жұмыс істесе, онда бір процесс аясында әртүрлі процессорларда бірнеше ағындар бір уақытта орындалуы мүмкін. Процестің ағынды дизайны бір процессорлы машиналарда да пайдалы. Бұл бірнеше логикалық әр түрлі

функцияларды орындайтын Бағдарламаның құрылымын жеңілдетуге көмектеседі.

Ағындар, процестер сияқты, орындалу күйлерімен сипатталады сонымен қатар, оларды бір-бірімен синхрондауға болады. Ағындардың негізгі күйлері, процестер сияқты, ағынның орындалу күйі, дайындық күйі және құлыптау күйі болып табылады. Жалпы алғанда, тоқтата тұру күйін ағындармен байланыстырудың қажеті жоқ, өйткені мұндай күйлерді процестер деңгейінде қарастыру қисынды. Атап айтқанда, егер процесс тоқтатылса, оның барлық ағындары міндетті түрде тоқтатылады, өйткені олардың барлығы осы процестің мекен-жай кеңістігін бөліседі. Ағын аяқталғаннан кейін оның регистрлері мен стектері жойылады.

5.2. Ағындардың түрлері

Әдетте ағындардың екі жалпы категорияға бөлінеді: пайдаланушы ағындары немесе пайдаланушы деңгейіндегі ағындар (user-level threads - ULT) және ядро деңгейіндегі ағындар (kemel - level threads-KLT). Әдебиетте екінші типтегі ағындар кейде ядро немесе жеңіл процестер (lightweight processes) қолдайтын ағындар деп аталады.

Пайдаланушы ағындары. Толығымен ULT ағындарынан тұратын бағдарламада барлық ағындарды басқару әрекеттерін қосымшаның өзі орындайды; ядро іс жүзінде ағындардың бар екенін білмейді. Қосымшаны көп ағынды ету үшін оны ядро деңгейіндегі ағындармен жұмыс істеуге арналған бағдарламалар жиынтығы болып табылатын арнайы кітапхананың көмегімен жасау керек. Ағындармен жұмыс істеуге арналған мұндай кітапханада ағындарды құруға және жоюға, ағындар арасында хабарламалар мен мәліметтер алмасуға, олардың орындалуын жоспарлауға, сонымен қатар олардың контекстін сақтауға және қалпына келтіруге болатын код бар. Алдыңғы абзацтарда сипатталған барлық оқиғалар бір процестің бөлігі ретінде пайдаланушы кеңістігінде болады. Ядро бұл әрекетті білмейді.

Ядро деңгейіндегі ағындар. Жұмысы толығымен ядро деңгейінде жұмыс істейтін ағындарға негізделген бағдарламада ағынды басқарудың барлық әрекеттері ядро арқылы жүзеге асырылады. Қолданбалар аймағында ағындарды басқаруға арналған код жоқ. Оның орнына ағындарды басқаратын ядро құралдарының қолданбалы бағдарламалау интерфейсі (application programming interface-API) қолданылады. Бұл тәсілдің мысалы-Windows операциялық жүйесі.

5.3. Көп ядролы және көп ағынды

Бірнеше ағыны бар бір қолданбаны қолдау үшін көп ядролы жүйелерді пайдалану өнімділік пен қолданбаны жобалау мәселелерін енгізеді.

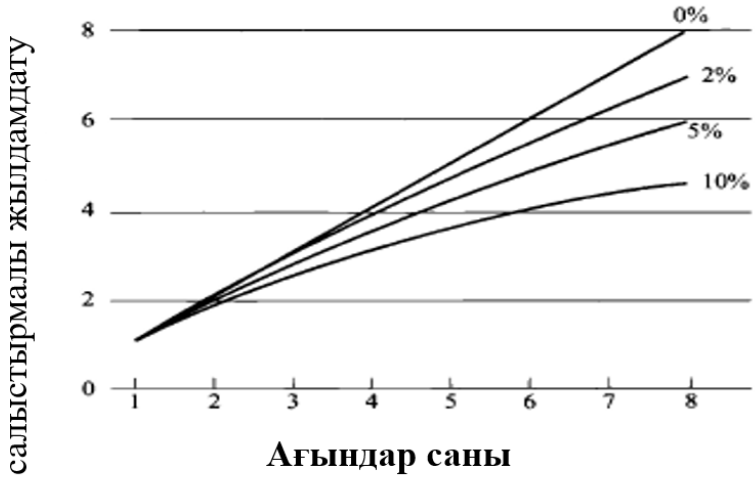
Көп ядролы жүйелердегі бағдарламалық қамтамасыз ету өнімділігі. Көп ядролы жүйелердің әлеуетті өнімділік артықшылықтары қолданбаға қолжетімді параллель ресурстарды тиімді пайдалану мүмкіндігіне

байланысты. Көп ядролы жүйеде жұмыс істейтін бір қолданбаны қарастырайық. Амдал заңында (4.1) [1]:

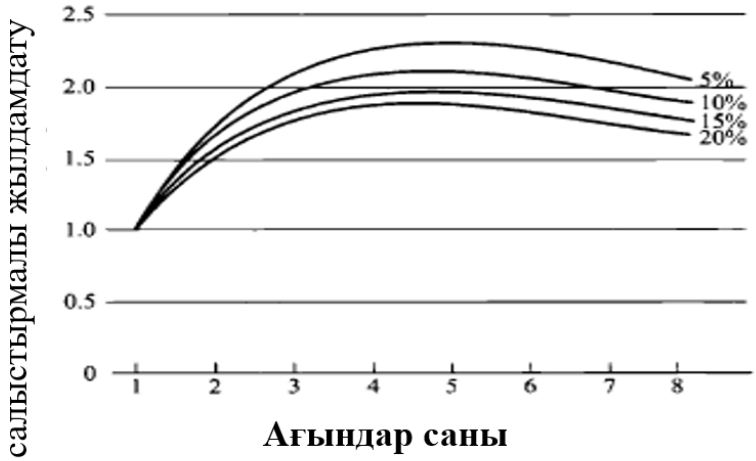
$$\text{Ускорение} = \frac{\text{Время выполнения программы на одном процессоре}}{\text{Время выполнения программы на } N \text{ параллельных процессорах}} == \frac{1}{(1-f)+\frac{f}{N}}$$

5.1

Заңда бағдарламаның орындалу уақытының бір бөлігі (1-f) дәйекті түрде орындалуы керек кодты орындауға жұмсалады, ал f бөлігі жоспарлауға байланысты үстеме шығындарсыз шексіз параллельге айналуы мүмкін код болып табылады. Бұл заң көп ядролы жүйелерді өте тартымды перспективаға айналдыруы керек. Бірақ 5.3 а суретінде көрсетілгендей, параллельденбеген кодтың аз мөлшері де бағдарламаның үдеуіне айтарлықтай әсер етеді. Егер кодтың тек 10% - ы (f=0,9) параллель болмаса, онда сегіз процессоры бар жүйеде бағдарламаны орындау тек 4,7 есе өнімділікті арттырады. Сонымен қатар, әдетте ақпарат алмасуға және бірнеше процессорлар арасында жұмысты бөлуге байланысты үстеме шығындар, сондай-ақ кэш консистенциясының үстеме шығындары бар. Мұның бәрі өнімділіктің шыңы болатын қисыққа әкеледі, содан кейін бірнеше процессорларды қолдануға байланысты үстеме шығындардың артуына байланысты оның нашарлауын көрсетеді (сурет. 5.3, б).



а) Әртүрлі реттік код көлемі үшін жылдамдату



б) Үстеме жүктememен үдеу

4.5-сурет. Есептеу өнімділігінің ядролар санына тәуелділігі

Дегенмен, бағдарламалық жасақтаманы әзірлеушілер бұл мәселелерді шешуде және көп ядролы жүйелердің қуатын тиімді пайдаланатын көптеген қолданбалар бар. Мәліметтер қорын басқару жүйелері және дерекқор қолданбалары көп ядролы жүйелерді тиімді пайдалануға болатын салалардың бірі болып табылады. Серверлердің көптеген түрлері параллельді көп ядролы архитектураларды жақсы пайдалана алады, өйткені әдетте серверлер параллельді өңдеуге болатын көптеген салыстырмалы тәуелсіз операциялармен жұмыс істейді. Жалпы мақсаттағы серверлік бағдарламалық қамтамасыз етуден басқа, өткізу қабілетін көбірек өзектермен масштабтау мүмкіндігінің пайдасын көретін қолданбалардың бірқатар сыныптары бар: жергілікті көп ағынды қолданбалар, олар ағындардың үлкен саны бар процестердің аз санымен сипатталады. ; бір ағынды процестердің үлкен санымен сипатталатын көппроцессті қолданбалар; қолданбаның бірнеше данасын бір уақытта іске қосу арқылы көп ядролы архитектурадан пайда алуға болатын көп даналық қолданбалар.

5.4 Windows жүйесінде процестер мен ағындарды басқару

Windows процестері. Қолданба бір немесе бірнеше процестерден тұрады. Әрбір процесс бағдарламаны орындауға қажетті ресурстарды қамтамасыз етеді. Әрбір процесс бір ағынмен басталады, көбінесе негізгі немесе негізгі ағын деп аталады, бірақ оның кез келген ағыны қосымша ағындарды жасай алады. Ағын - іске қосуды жоспарлауға болатын процесс ішіндегі нысан.

Тапсырма нысаны процестерді бір нысан ретінде басқару үшін біріктіруге мүмкіндік береді. Жұмыс нысандары - олармен байланысты процестердің атрибуттарын басқаратын, қорғалатын, ортақ нысандар.

Ағын пулі – бұл қосымшаның атынан асинхронды кері қоңырауларды тиімді орындайтын жұмыс ағындарының жиынтығы. Ағындық бассейн бағдарлама ағындарының санын азайту және жұмыс ағынын басқару үшін қолданылады.

Талшық (fiber) – бұл қолданба қолмен жоспарлауы керек орындалу бірлігі. Талшықтар оларды жоспарлаған ағындар аясында жасалады. Әр ағын бірнеше талшықты жоспарлай алады. Жүйе тұрғысынан талшық оны жүзеге асыратын ағынға ұқсас.

Windows процестерінің маңызды сипаттамалары келесідей:

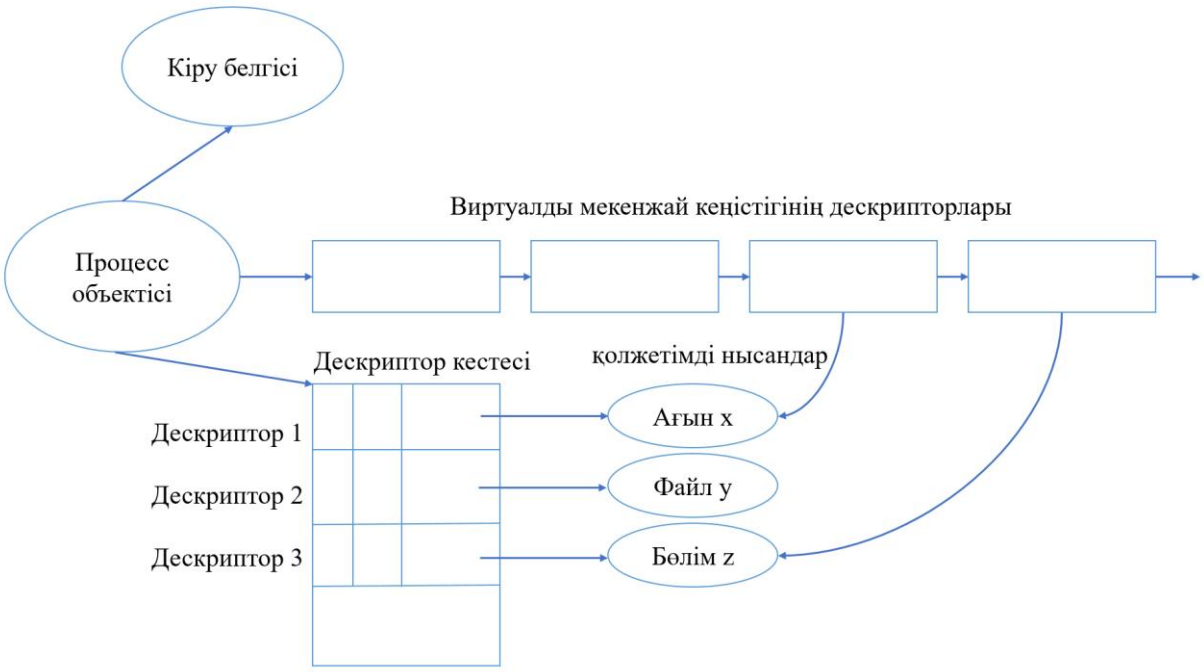
− Windows процестері объектілер ретінде жүзеге асырылады.

− Процесті жаңа процесс ретінде немесе бұрыннан бардың көшірмесі ретінде жасауға болады.

− Орындалатын процесс бір немесе бірнеше ағынды қамтуы мүмкін.

− Процесс нысандарында да, ағын нысандарында да кірістірілген синхрондау мүмкіндіктері бар.

Суретте. 5.4 ([212] негізінде) процесс пен процесс басқаратын немесе пайдаланатын ресурстар арасындағы байланысты көрсетеді. Қауіпсіздік мақсатында әрбір процеске негізгі таңбалауыш немесе процесс таңбалауышы деп аталатын қатынас белгісі тағайындалады. Процестер сонымен қатар қазіргі уақытта процеске тағайындалған виртуалды мекенжай кеңістігін анықтайтын бірқатар блоктармен байланысты. Процесс бұл құрылымдарды тікелей өзгерте алмайды; бұл жағдайда ол процеске жадты бөлу қызметін қамтамасыз ететін виртуалды жад менеджеріне сүйену керек. Тиісті қорғалған ішкі жүйеге кіретін операциялық жүйе қолданбасының сұранысы бойынша процесс жасалады. Ішкі жүйе өз кезегінде процесті құруға сұранысты Windows орындаушысына жібереді, ол процесс объектісін жасайды және оның дескрипторын ішкі жүйеге қайтарады. Процесті құру кезінде Windows операциялық жүйесі автоматты түрде ағынды жасамайды. Осылайша, Win32 ішкі жүйесі жаңа процесс ағынын жасау және оған дескриптор алу үшін Windows процесс менеджерін қайта қарап шығады.



5.4-сурет. Windows процестері және олардың ресурстары

Процестер мен ағындардың объектілері. Windows операциялық жүйесінің объектіге бағытталған құрылымы процестермен жұмыс істеу үшін жалпы мақсаттағы ішкі жүйені құруды жеңілдетеді. Windows процестерге байланысты нысандардың екі түрін қолданады: процестер мен ағындар. Процесс-бұл жад және ашық файлдар сияқты өз ресурстарына ие пайдаланушының тапсырмасына немесе қосымшасына сәйкес келетін объект. Ағын-бұл басқарылатын жұмыс бірлігі, ол дәйекті түрде орындалады және процессорға басқа ағынның орындалуына ауысуға мүмкіндік береді. Windows амалдық жүйесіндегі әрбір процесс объектімен ұсынылған.

Процестің әр нысаны белгілі бір атрибуттарды қамтиды және бірқатар әрекеттерді немесе ол орындай алатын қызметтерді инкапсуляциялайды. Процесс бірнеше жарияланған интерфейс әдістері арқылы шақырылған кезде әрекеттерді орындайды. Жаңа процесті құру кезінде Windows амалдық жүйесі Windows процесі нысандарының жаңа даналарын құру үшін процесс үлгісі ретінде анықталған объектінің класын немесе түрін қолданады. Нысанды құру кезінде оның атрибуттарына нақты мәндер беріледі.

Орындалғанға дейін Windows процесінде кем дегенде бір ағын болуы керек, содан кейін ол басқа ағындарды жасай алады. Windows-та жасалған ағын алты күйдің бірінде болуы мүмкін (5.5-сурет). Мультипроцессорлық жүйеде бір процестің бірнеше ағындары параллель орындалуы мүмкін. Windows амалдық жүйесі процестердің параллель орындалуын қолдайды, өйткені әртүрлі процестердің ағындары параллель орындалуы мүмкін. Сонымен қатар, бір процестің бірнеше ағынына әртүрлі процессорларды

бөлуге болады және бұл ағындарды бір уақытта да жасауға болады. Параллелизмге бірнеше процестерді қолдануға жұмсалатын шығынсыз көп ағынды процесте қол жеткізіледі. Бір процестің ағындары ортақ мекен-жай кеңістігін қолдана отырып, бір-бірімен ақпарат алмасады және процестің бірлескен ресурстарына қол жеткізе алады. Әр түрлі процестерге жататын ағындар осы екі процесс үшін орнатылған Жалпы жад аймағын қолдана отырып, бір-бірімен ақпарат алмасуы мүмкін.

Сурет 5.5. Windows амалдық жүйесіндегі ағындардың күйлері

Нысанға бағытталған көп ағынды процесс серверлік қосымшаларды іске асырудың тиімді құралы болып табылады. Мысалы, бір қызмет көрсету процесі бірнеше клиенттерге қызмет көрсете алады. Клиенттің әр сұранысы серверде жаңа ағын құруға әкеледі.

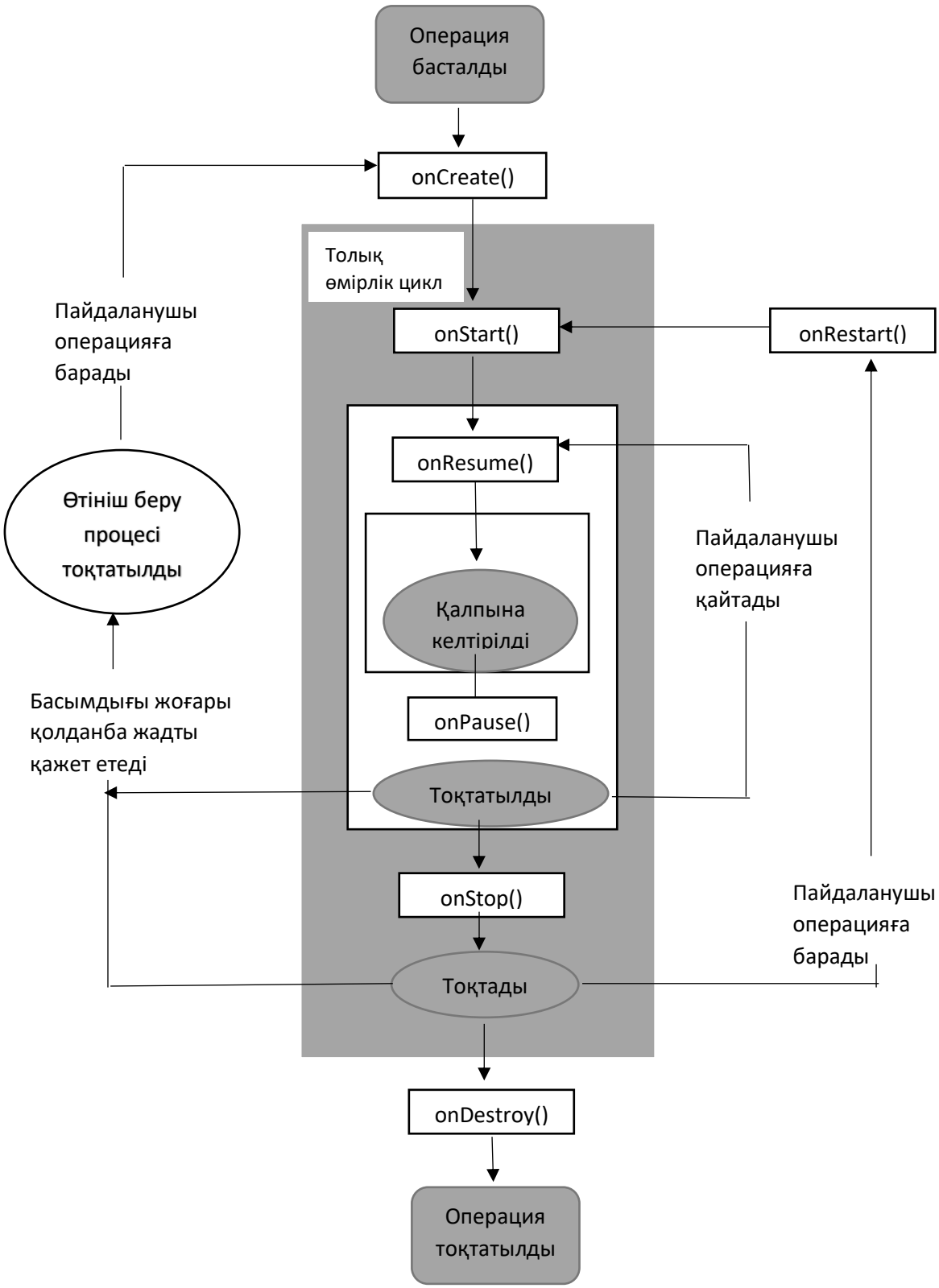
5.5 Android-де процестер мен ағындарды басқару.

Android Қосымшалары. Android қосымшасы-бұл тиісті функционалдылықты жүзеге асыратын бағдарламалық жасақтама. Әрбір Android қосымшасы бағдарлама компоненттерінің төрт түрінің бір немесе бірнеше данасынан тұрады. Әрбір компонент тұтастай алғанда қосымшаның мінез-құлқында белгілі бір рөл атқарады және әр компонентті қосымшада,

тіпті басқа қосымшаларда да қосуға болады. Төменде компоненттердің төрт түрі берілген.

1. Операциялар (әрекеттер). Операция UI ретінде көрінетін бір экранға сәйкес келеді. Мысалы, электрондық пошта қосымшасында жаңа электрондық пошта хабарламаларының тізімін көрсететін бір операция болуы мүмкін, ал басқа операция - электрондық пошта хабарламасын құрастыру, ал екіншісі-электрондық поштаны оқу. Операциялар электрондық поштамен жұмыс істеуге арналған бірыңғай қосымшаны құру үшін бірге жұмыс істесе де, олардың әрқайсысы басқаларына тәуелсіз.
2. Қызметі (services). Қызметтер, әдетте, аяқтауға көп уақытты қажет ететін фондық операцияларды орындау үшін қолданылады. Бұл Пайдаланушы тікелей өзара әрекеттесетін қосымшаның негізгі ағынына жауап берудің аз уақытын қамтамасыз етеді (сонымен қатар пайдаланушы интерфейсінің ағымы). Мысалы, қызмет пайдаланушы басқа бағдарламада болған кезде музыканы фонда ойнату үшін ағын жасай алады немесе пайдаланушының операциямен өзара әрекеттесуін бұғаттамай желі арқылы деректерді алу үшін ағын жасай алады.
3. Мазмұн провайдерлері (content provider). Мазмұн провайдері қолданба пайдалана алатын қолданба деректерінің интерфейсі ретінде әрекет етеді. Басқарылатын деректер санаттарының бірі-мазмұн провайдері бар қосымшада ғана қолданылатын жабық деректер.
4. Хабар таратушылар (broadcast receiver). Хабар тарату хабарламаларын алушы жүйе деңгейіндегі барлық хабарландыруларға жауап береді.

Операция-бұл пайдаланушылармен өзара әрекеттесу экранын қамтамасыз ететін бағдарлама компоненті. - Сур. 5.6 операция күйлерінің ауысу диаграммасының жеңілдетілген көрінісін көрсетеді.



Сурет 5.6. Операция күйлерінің ауысу диаграммасы

Android процестері мен ағындары. Әдепкі бойынша, бағдарлама жалғыз процесс пен жалғыз ағынды ажыратады. Қосымшаның барлық компоненттері

осы қосымшаның жалғыз процесінің жалғыз ағынында орындалады. Компонент баяу немесе блоктау әрекеттерін орындау кезінде пайдаланушы интерфейсінің баяулауын болдырмас үшін, әзірлеуші процесс аясында және/немесе қолданба ішінде бірнеше процестер жасай алады. Қалай болғанда да, осы қосымшаның барлық процестері және олардың ағындары бірдей виртуалды машинада орындалады.

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