

Дәріс №5. Тақырыбы: Процесс туралы түсінік. Процестерді басқару. Процестердің жағдайы.

Операциялық жүйенің негізгі функциясы жүйенің аппараттық және программалық ресурстарының рациональді пайдаланылуы болып табылады. Негізгі ресурстарға: процессорлар, жады, сыртқы құрылғылар, мәліметтер және программалар. Есептеу машинасының жұмысына тікелей әсер ететін операциялық жүйенің негізгі бөлігі процестерді басқаратын ішкі жүйе болып табылады.

Процесс (немесе басқаша есеп) – программаны немесе операциялық жүйенің қандайда бір процедураларды орындауға бөлінген жадының виртуальді адресілік кеңістігі.

Операциялық жүйе үшін процесс жұмыс бірлігі болып табылады. Процестерді басқарудың ішкіжүйесі процестердің орындалуын жоспарлайды, яғни процессорлық уақытты біртебірте болатын жүйедегі процестердің арасында бөледі, сонымен қатар процестерді құрады және жояды, процестерді қажетті жүйелік қорлармен (ресурстармен) қамтамасыз етеді, процестер арасындағы байланысты қамтамасыз етеді.

Ядро — компьютердің ресурстарына қосымша программалардың қол жеткізуін қамтамасыз ететін, ОЖ орталық бөлігі. Мысалы, процессорлық уақыт, жедел жады, сыртқы құрылғылар. Әдетте файлдық жүйенің қызметінен көрсетеді.

ОЖ ядросында келесі функцияларды орындау үшін арнайы программалар бар:

Үзулерді өңдеу;

Процестерді құру және жою;

Процестердің бір жағдайдан екіншісіне өтуі;

Процестер арасындағы байланыстарды ұйымдастыру, соның ішінде процестерді синхрондау (уақыт үйлесімдендіру)

Енгізу шығару амалдарын орындау;

Жадыны үлестіруді және қайта үлестіруді ұйымдастыру;

Файлдық жүйенің жұмысын қамтамасыз ету;

ЭЕМ және ЕЖ жұмысына байланысты тіркеу жүргізу;

Монолиттік ядро

Монолиттік ядро – бұл ОЖ ұйымдастыру сұлбасы, ол кезде оның барлық компоненттері бір программаның құрамдас бөліктері болып табылады, мәліметтердің жалпы құрамдарын пайдаланады және өзара процедураларды тікелей шақыру арқылы байланысады.

Монолиттік ядроның барлық бөліктері бір адресілік кеңістікте жұмыс істейді.

Артықшылықтары: Жұмыстың жылдамдығы, модулдердің жинақталған дайындалуы.

Кемшіліктері: ядро толығымен бір адресілік кеңістікте жұмыс істегеннен кейін, оның бір бөлігінің жұмысының бұзылуы бүкіл жүйенің жұмысына әсерін тигізеді.

Монолиттік ОЖ үшін ядро бүтіндей жүйемен сәйкес болады.

Мысалдары: UNIX ОЖ ядролары, Linux; MS-DOS ядросы

Модульдік ядро — компьютерлердің ОЖ-нің монолиттік ядроларының қазіргі заманға сай жетілдірілген түрі. Монолиттік ядролардан айырмашылығы, модульдік ядролар компьютердің аппараттық жабдықталуының құрамы өзгерген кезде қайта компиляциялауды талап етпейді. Оның орнына модульдік ядролар басқа механизм ұсынады, мысалы, драйверлерді.

Мысалы, VFS — «виртуальді файлдық жүйе», Linux ядросында файлдық жүйелердің көптеген модулдерімен бірге пайдаланылады.

Микроядро процестерді басқарудың элементар функцияларын және құрылғылармен жұмыс істеудің абстракцияланған шағын жиынын ұсынады. Жұмыстардың көптеген бөліктері қызметтер (сервистер) деп аталатын арнайы қолданушылардың процестері арқылы орындалады.

Артықшылықтары: жүйе бөліктеріндегі қателерге және құрылғылардағы ақауларға орнықтылық.

Кемшіліктері: процестер арасындағы мәліметтерді алмасу қосымша шығындарды талап етеді.

Микроядро программалардың арасындағы өзара байланыстарды, процессорды пайдалануды жоспарлауды, үзілістерді өңдеуді, жадыны басқарауды қамтамасыз етеді.

Мысалдар: QNX; Mach, AIX; Minix ; ChorusOS ; AmigaOS; MorphOS.

Экзоядро — компьютерлердің ОЖ ядросы, бұдар тек процестердің байланысу функциясын және ресурстарды қауіпсіз бөлуге және босатуға негізделеді.

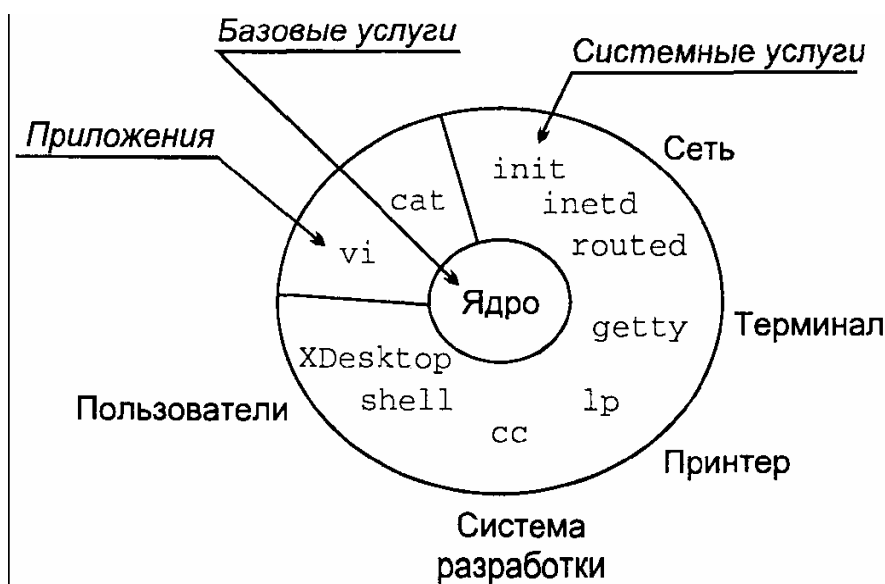
Наноядро — бұл компьютерлердің ОЖ ядросының архитектурасы, оның барысында ядро тек бір ғана есепті —компьютер құрылғыларындағы аппараттық үзулерін өңдеу.

Үзулерді өңдеуден кейін наноядро өңдеу нәтижелері туралы ақпараттарды жоғары орналасқан программалық жабдықтауға ұзу механизмі арқылы жөнелтеді.

Мысалы: KeyKOS – ең алғашқы наноядро негізіндегі ОЖ. Алғашқы нұсқасы 1983-жылы пайда болды.

Гибридтік ядро микроядроның түрлендірілуі.

Микроядролық архитектураның элементтері және монолитті ядроның элементтері WindowsNT ядросына топтастырылған. Сондықтан Windows NT ОЖ гибридтік ОЖ.



Ядро келесі 3 негізгі бөліктерден тұрады:

1. Файлдік жүйелер;
2. Процестерді және жадыны басқарудың ішкі жүйесі;
3. Енгізу-шығарудың ішкі жүйесі.

ОЖ есептеу ресурстарына процестердің қол жеткізуін басқарады және бірнеше тапсырмалардың қатар орындалып жатқанындай көрсетеді.

Ядроның негізгі тапсырмасы, процестердің жоспарлаушы, процестердің арасындағы жүйелік ресурстар үшін бәсекелестігін шешеді. Жоспарлаушы процестерді іске қосады, оның бөлінетін ЖР жеке алып алмауын қадағалайды. Приоритеті жоғары процесті таңдайды және іске қосады.

Орталық процессор (ОП) немесе орталық процессорлық

құрылғы (ОПҚ) (ағыл. *central processing unit* — **CPU**) — машиналық нұсқаулардың процессоры, есептеу процесіндегі ақпараттардың өңделуіне жауап береді. Қазіргі кездегі ОПҚ жеке микросхемалар түрінде (чиптер) дайындалады, оларды микропроцессорлар деп атайды. Қарастырылған мәселелер ядроның архитектурасына және оны іске асыруға байланысты әртүрлі болады.

Процестердің жағдайлары

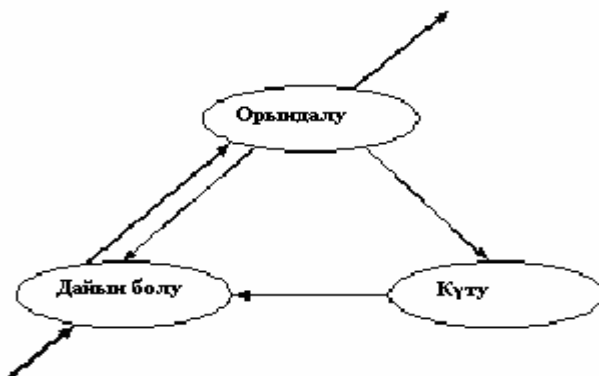
Көпміндетті (көппроцессорлы) жүйеде процесс келесі 3 жағдайдың біреуінде бола алады:

ОРЫНДАЛУ – процестің белсенді жағдайы, оның барысында процесс қажетті ресурстарға ие және тікелей процессорда орындалады;

КҮТУ – процестің белсенді емес күйі, процесс оқшауланған (заблокирован), ішкі жағдайларға байланысты ол орындала алмайды, белгілі бір жағдайдың орындалуын күтеді, мысалы, енгізу-шығару амалдарын аяқтау, басқа процестен хабарлама алу, өзіне қажетті қандайда бір ресурсты босатады.

ДАЙЫН БОЛУ – процестің белсенді емес жағдайы, бұл кезде процесс сыртқы жағдайлардың әсерінен оқшауланған: процесс өзіне қажет барлық ресурстарға (қорларға) ие, ол орындалуға дайын, бірақ процессор басқа процестің орындалуымен бос емес.

Процестерді жоспарлау алгоритміне сәйкес өмір сүру барысында процесс бір жағдайдан басқа жағдайға ауысады, берілген операциялық жүйеде. Процестің графы келесі суретте көрсетілген:



Бір процессорлы жүйеде ОРЫНДАЛУ жағдайында тек бір ғана процесс бола алады, ал әр КҮТУ және ДАЙЫН БОЛУ жағдайларында – бірнеше процестер болып, осы процестер күтетін және дайын процестердің кезегін құрайды. Процестің өмір циклі ДАЙЫН БОЛУ жағдайынан басталады, процесс орындалуға дайын және өзінің кезегін күткен кезде. Процесс белсенді болғанда ОРЫНДАЛУ жағдайын көшеді және онда процессорды өзі босатып, КҮТУ жағдайына өтеді, немесе процессордан күштеп шығарылып тасталады, мысалы, берілген процеске бөлінген процессорлық уақыт квантының бітуіне байланысты. Соңғы жағдайда процесс ДАЙЫН БОЛУ жағдайына қайтарылады. Күтілетін жағдай болған кезде осы жағдайға процесс КҮТУ жағдайынан ауысады. Процестің өмір сүру бойында оның орындалуы үзіліп және жалғастырылады. Процестің қайта орындалуы үшін оның жағдайын операциялық ортадан қалпына келтіруге болады. Операциялық жүйенің жағдайы регистрлерінің жағдайы және программалық санауыштың, процессордың жұмыс режимі, ашық файлдарға сілтемелер, аяқталмаған енгізу-шығару операциялары туралы ақпараттар, берілген процестің орындалу барысындағы жүйелік шақырулардың қателер коды және т.с.с. арқылы көрсетіледі.

Бұл ақпарат процестің **контексі** деп аталады. Сонымен қатар, операциялық жүйеге процесті жоспарлау үшін қосымша ақпараттар қажет: процестің идентификаторы, жағдайы, процестің артықшылық деңгейлері туралы мәліметтер, кодалық сегменттің орналасуы және басқа ақпарат. ОЖ-да процестерді жоспарлау үшін пайдаланылатын осы

түрдегі ақпарат процестің **дескрипторы** деп аталады. Процестерді жоспарлаудың ішкіжүйесіне жеңіл қатынас болатын процестің дескрипторы контекспен салыстырғанда өте жедел ақпарат сақтайды. Процестің контексті маңызды емес ақпарат сақтайды, операциялық жүйеде үзілген процесті қалпына келтіру туралы шешім қабылданғанда пайдаланылады.

Программалық кода орындалып басталады, егер операциялық жүйеде оған арналған процесс құрылса. Процесс құру дегеніміз:

1. берілген процессті сипаттайтын, ақпараттық құрылымдарды (дескриптор және контекст) құру үшін.
2. дайын процестердің кезегіне жана процесті қосу үшін дескриптор енгізіледі.
3. процестің кодалық сегментін жедел жадыға жүктеу немесе свопинг аймағына.

Процестерді жоспарлау алгоритмі

Процестерді жоспарлау келесі мәселелерді шешуді қарастырады:

1. орындалып жатқан процестерді ауыстыру үшін уақыт кезеңін анықтау.
2. дайын процестердің кезегінен орындауға процестерді таңдау.
3. «жаңа» және «ескі» процестерді ауыстыру.

Алдыңғы екі есеп программалық құралдар арқылы шешіледі, ал соңғысы аппаратты түрде.

Жиі кездесетін алгоритмдердің екі түрі болады: кванттауға негізделген алгоритмдер және приоритеттерге негізделген алгоритмдер.

Кванттауға негізделген алгоритмдерге сәйкес, әр процеске оны процессорда орындауға кететін кванттық уақыты бөлінеді және қарқынды процесс ауысады, егер

- процесс аяқталып және жүйеден кетсе;
- қате кездессе;
- процесс КҮТУ жағдайына көшсе;
- берілген процеске бөлінген процессорлық уақыт кванты аяқталса.

Өзіне бөлінген уақыт квантын аяқтаған процесс ДАЙЫН жағдайына көшеді және өзіне жаңа кванттық уақыт берілетінін күтеді, ал орындалуға дайын тұрған процестердің кезегінен белгілі бір ережеге сай таңдалады. Сонымен, бірде –бір процесс процессорды ұзақ уақыт ұстамайды сондықтан кванттау уақыт бөлу жүйелерінде пайдаланылады.

Процесс, который исчерпал свой квант, переводится в состояние ГОТОВНОСТЬ и ожидает, когда ему будет предоставлен новый квант процессорного времени, а на выполнение в соответствии с определенным правилом выбирается новый процесс из очереди готовых. Таким образом, ни один процесс не занимает процессор надолго, поэтому квантование широко используется в системах разделения времени. Граф состояний процесса, изображенный на рисунке 2. соответствует алгоритму планирования, основанному на квантовании.

Кванты, выделяемые процессам, могут быть одинаковыми для всех процессов или различными. Кванты, выделяемые одному процессу, могут быть фиксированной величины или изменяться в разные периоды жизни процесса. Процессы, которые не полностью использовали выделенный им квант (например, из-за ухода на выполнение операций ввода-вывода), могут получить или не получить компенсацию в виде привилегий при последующем обслуживании. По разному может быть организована очередь готовых

процессов: циклически, по правилу "первый пришел - первый обслужился" (FIFO) или по правилу "последний пришел - первый обслужился" (LIFO).