

КӨПАҒЫНДЫҚ ПРОГРАММАЛАУДЫ ҚОЛДАУ. ҮРДІСТЕРДІ БАСҚАРУ. ЖАДЫНЫҢ ҰЙЫМДАСТЫРЫЛУЫ.

«ПРОГРАММАЛАУ ТЕХНОЛОГИЯЛАРЫ» ПӘНІ

ЛЕКТОР: АБДРАХМАНОВА М.Б.

ПАРАЛЛЕЛЬ КОМПЬЮТЕРЛЕРДІҢ ЖІКТЕЛУІ

- Параллель компьютерлердің ең танымал классификациясын Флинн ұсынған, ол компьютерлер жүзеге асыратын параллелизм формасын көрсетеді. Жіктеудің негізгі ұғымдары «командалар ағыны» және «мәліметтер ағыны» болып табылады. Командалар ағынын бір программаның командалар тізбегі ретінде қарастыруға болады. Мәліметтер ағыны - бұл бір программа өңдейтін мәліметтердің тізбегі. Бұл классификацияға сәйкес компьютерлердің төрт үлкен класы бар:
- 1) SISD (Single Instruction - Single Data): жалғыз командалар ағыны - жалғыз мәліметтер ағыны. Бұл тізбекті компьютерлер, онда бір программа орындалады, яғни тек бір командалар санауышы болады.
- 2) SIMD (Single Instruction – Multiple Data): жалғыз командалар ағыны - көптік мәліметтер ағыны). Мұндай компьютерлерде жалғыз программа орындалады, бірақ әрбір команда мәліметтер жиынын өңдейді. Бұл параллелизмнің векторлық түріне сәйкес келеді.
- 3) MISD (Multiple Instruction - Single Data) – көптік командалар ағыны – жалғыз мәліметтер ағыны. Бұл жағдайда бір уақытта бірнеше команда бір мәліметтер элементімен жұмыс істейді деп есептеледі (іс жүзінде қолданылмаған).
- 4) MIMD (Multiple Instruction - Multiple Data) - (бірнеше командалық ағын - бірнеше мәліметтер ағыны). Мұндай компьютерлерде бір уақытта бір-бірінен тәуелсіз түрде бірнеше программалық тармақтар орындалады, олар белгілі бір уақыт аралықтарында мәліметтермен алмасады. Мұндай жүйелерді әдетте көппроцессорлы жүйелер деп атайды.

КӨППРОЦЕССОРЛЫ ЖҮЙЕЛЕР

- MIMD ЭЕМ екі түрі бар: ортақ жадылы (с разделяемой (общей)) және жеке (распределенной (индивидуальной)) жадылы.
- Ортақ жадылы:
 - Процессораралық мәлімет алмасу уақыты үнемделеді;
 - Программалау тәсілін жеңілдетеді;
 - Синхронизация құралдары қажет (ОС жұмыс механизмін күрделендіреді);
 - Жады коммутаторының өткізгіштігі жоғары болуы тиіс.
- Жеке жадылы:
 - Тәуелсіз адрестік кеңістік;
 - Процессорлар арасындағы мәліметтердің физикалық көшірілуі;
 - Жады коммутаторына талап төмен;
 - Процессор санына шектеу төмен.

АМДАЛ ЗАҢЫ

- Параллель жүйелердің негізгі сипаттамаларының бірі - параллель жүйенің R үдеуі: $R = T1 / Tn$,
 $T1$ - бірпроцессорлы жүйеде есепті шығару уақыты, Tn - n -процессорлық жүйеде сол мәселені шешу уақыты.
- Амдал заңы: $R = W * t / (W_{ск} + W_{пр} / n) \rightarrow_{n \rightarrow \infty} 1 / a$
- $W = W_{ск} + W_{пр}$ болсын, мұндағы W - есептегі амалдардың жалпы саны, $W_{пр}$ - параллель орындалатын амалдар саны, $W_{ск}$ - скалярлық (параллелденбейтін) амалдар саны. Сонымен қатар біз бір операцияның орындалу уақытын t арқылы белгілейміз. Мұндағы $a = W_{ск} / W$ - скалярлық амалдардың үлесі.
- Амдал заңы параллель есептеулер үшін принципиалды маңызды позицияларды анықтайды:
- 1. Үдеу тапсырманың потенциалды параллелдігіне (мәні $1 - a$) және аппараттық параметрлерге (процессорлар саны n) байланысты.
- 2. Шектік үдеу есептің қасиеттерімен анықталады. Мысалы, $a = 0,2$ болсын (бұл нақты мән), онда үдеу кез-келген процессор саны үшін 5-тен аспауы керек, яғни максималды үдеу тапсырманың потенциалды параллелизмімен анықталады.
- Амдал заңының негізгі нұсқасы процессорлар арасында хабарлама алмасуға жұмсалатын уақытты ескермейді.

ЖЕЛІЛІК АМДАЛ ЗАҢЫ

- $R_c = W_t / ((W_{ск} + W_{пр} / n) t + W_c * t_c) = 1 / (a + (1 - a) / n + W_c t_c / W_t) = 1 / (a + (1 - a) / n + c)$
- Мұнда W_c - мәліметтерді жіберу саны, t_c - мәліметті бір реттік жіберу уақыты.
- Амдалдың желілік заңы: $R_c = 1 / (a + (1 - a) / n + c)$
- Бұл заң көппроцессорлы есептеудің келесі екі ерекшеліктерін анықтайды:
- 1. c мәні - есептеудің желілік деградация коэффициенті:
- $c = W_c t_c / W_t = c_A c_T$ - бір реттік мәлімет жіберуге кететін есептеу көлемін анықтайды (жұмсалған уақыт бойынша).
- c_A - деградация коэффициентінің алгоритмдік компонентін, ал c_T - техникалық компонент.
- Осылайша, есептеу жылдамдығын арттыру үшін деградация коэффициентінің екі компонентіне де әсер ету қажет.
- 2. Тапсырма мінсіз параллелизмге ие болса да, желілік үдеу:
- $R_c = 1 / (1 / n + c) = n / (1 + c n) \rightarrow_{c \rightarrow 0} n$

MIMD КЛАСТАРЫ (ҚҰРЫЛЫМДЫҚ-ФУНКЦИОНАЛДЫ АСПЕКТ БОЙЫНША)

- Симметриялық мультипроцессорлар (SMP) – ортақ жады, біртекті процессорлар, процессорлар жадыға шина немесе коммутатор көмегімен қосылады, ортақ жалғыз ОЖ, процессорлар санына шектеу
- Жаппай параллелизм жүйелері (MPP) – жеке жады, процессор саны көп, толық ОЖ бір түйінде (қалған түйіндерде қысқартылған нұсқасы) немесе әрбірінде жеке толық ОЖ, түйіндер коммуникациялық орта арқылы байланыстырылады, хабарламалар жөнелту принципі
- Кластерлер – MPP жүйелердің бір түрі деуге болады, дайындығы жоғары компоненттерден құралады, әр түйіннің жеке ОЖ көшірмесі бар, түйіндердің мүмкіндіктері әркелкі болуы ықтимал.

ПРОЦЕСТЕР

- Процесс белсенді объект болып табылады, оның программалық санауышы (атқарылатын келесі команданы анықтайды) және байланысқан ресурстар жиыны бар. Процестің жады құрылымы бірнеше бөлімдерден тұрады:
- Мәтіндік бөлім – атқарылатын код – өлшемі бекітілген
- Мәліметтер бөлімі – ауқымды айнымалылар – өлшемі бекітілген
- Бөлім секциясы – программаның атқарылу барысында динамикалық түрде үлестірілетін жады
- Стек секциясы – функцияларды шақыру кезіндегі уақытша мәліметтер қоймасы (функция параметрлері, қайтару адресі, жергілікті айнымалылар сияқты мәліметтерді сақтайды)
- Әртүрлі процестер бір программамен байланысты болуы мүмкін, бірақ олардың әрқайсысы жеке атқару тізбегі болып есептеледі.
- Процесс атқарылу кезінде әртүрлі күйлерде болуы мүмкін (жаңа, іске қосылған, күту, дайын, аяқталды). Операциялық жүйеге байланысты процесс күйлерінің атаулары әртүрлі болады.
- Процессордың кез келген ядросында кез келген уақыт мезетінде тек бір процесс атқарыла алады. Бірақ күту күйіндегі процестер көп болуы мүмкін.
- Процесс операциялық жүйеде процесті басқару блогымен (PCB) бейнеленеді. Оның құрамында нақты процесспен байланысты бірқатар ақпарат болады.

PCB ҚҰРАМЫ

- Процестің күйі
- Программа санауышы – процесс үшін орындалуы тиіс келесі нұсқаудың адресін көрсетеді
- Процессор регистрлері – белсенді күйде процестің орындалуы үшін қажетті ақпаратты сақтайтын регистрлер жиыны
- Орталық процессор жоспарлауы туралы ақпарат – процесс басымдығы, жоспарлау кезектеріне нұсқауыштар және т.б.
- Жадыны басқару туралы ақпарат – базалық және шектік регистрлер, беттер кестелері
- Есептік ақпарат – процестің орындалуына жұмсалатын орталық процессор көлемі, уақыт бойынша шектеулер және т.б.
- Енгізу/шығару күйі туралы ақпарат – процеске бөлінген енгізу/шығару құрылғыларының тізімі, ашық файлдар тізімі, т.б.

ПРОЦЕСТІ ЖОСПАРЛАУ

- Мультипрограммалау қызметі – Орталық процессордың қолданылуын барынша арттыру. Орталық процессор ядросын процессорлар арасында белгілі бір уақыт интервалдарында ауыстыру арқылы қолданушыға әртүрлі программалармен байланысу мүмкіндігін береді. Бұл мақсат үшін процестер жоспарлаушысы ядрода программаны орындау үшін қолжетімді процесті таңдайды. Әрбір ядро бір уақытта бір ғана процесті іске қоса алады.
- Қазіргі уақытты жадыда орналасқан процестер саны - мультипрограммалау деңгейі.
- Процестердің басым бөлігі енгізу/шығарумен (есептеулерге қарағанда, енгізу/шығаруға көп уақыт жұмсайтын процесс) немесе есептеулермен байланысты болып сипатталады.

ПРОЦЕСТІ ЖОСПАРЛАУ

- Операциялық жүйенің үзілістерінің (прерывание) мақсаты – орталық процессор ядросын ағымдағы тапсырмадан босатып, ядроның ішкі программасын іске қосу. Үзіліс болған кезде жүйе ядроға іске қосылған процестің ағымдағы контекстін сақтау керек, себебі өңдеу аяқталғаннан кейін процестің контекстін қайта іске қосып, жалғастыру қажет. Контекстке орталық процессор регистрлерінің мәндері, процестің күйі, жадыны басқару туралы ақпарат кіреді.
- Контекстіні ауыстыру (переключение контекста) – ағымдағы процестің күйін сақтап, басқа процестің күйін қалпына келтіре отырып, орталық процессор ядросын жаңа процеске ауыстыру. Контекстіні ауыстыру уақыты аппараттық қолдауға тәуелді.

ПРОЦЕСТІ ЖОСПАРЛАУ

- Бір уақытта орындалатын процестер өзара тәуелсіз немесе өзара әрекеттесуші болуы мүмкін. Жүйеде орындалып жатқан басқа процестермен мәлімет алмаспайтын процесс тәуелсіз болып есептеледі. Өзара әрекеттесу келесі мақсаттарда орындалуы мүмкін: мәліметті ортақ пайдалану, есептеулерді жылдамдату (тапсырманы параллель орындалатын бөліктерге бөлу), модульдік (жүйелік функцияларды жеке процестер арасында бөлу).
- Процессаралық әрекеттесу модельдері: ортақ жады және хабарламалармен алмасу.

АҒЫН

- Ағын бір процестің аясында бірнеше тапсырманы бір уақытта орындауға мүмкіндік береді. процессорды жүктеудің базалық бірлігі; онда ағын идентификаторы, программалар санауышы, регистрлер жиыны және стек болады. Ол сол процеске тиесілі басқа ағындармен кодтың бөлімін, деректер бөлімін және ашық файлдар мен сигналдар сияқты операциялық жүйенің басқа да ресурстарын бөліседі. Дәстүрлі процестің бірыңғай басқару ағыны бар. Егер процестің бірнеше басқару ағыны болса, ол бір уақытта біреуден артық тапсырмаларды орындауы мүмкін.

КӨПАҒЫНДЫҚ ПРОГРАММАЛАУ АРТЫҚШЫЛЫҚТАРЫ

- Жауап қайтару жылдамдығы — қосымшаның бір бөлігі бұғатталған немесе ұзақ уақыттық операциямен жүктелген болса да, программа жұмысын жалғастыру.
- Ресурстарды ортақ пайдалану — ағындар процестің жадысы мен ресурстарын бөліп пайдалана алады.
- Үнемділік — процесс құру үшін жады мен ресурстар бөлу үлкен шығындарға алып келеді. Ал ағындар құру және контекстіні ауыстыру шығындары әлдеқайда аз.
- Масштабтау — ағындарды параллель орындау мүмкіндігі.

ПАРАЛЛЕЛИЗМ ТИПТЕРІ

- Мәліметтер параллельдігі – мәліметтердің ішкі жиындарын әртүрлі ядроларға жіктеп, әрбір ядроға бірдей операцияларды орындау.
- Тапсырмалар параллельдігі – әр ағын жеке тапсырманы орындайды.

ҰҒЫМДАР

- Айқын емес көпағындық – ағындарды құру және басқаруды қосымшаларды құрушылардан компиляторлар мен атқару уақытының кітапханаларына беру (тапсырмалар құру арқылы).
- Ағындар пулы – іске қосу кезінде бірнеше ағынды құру және оларды пулға орналастыру, пулда ағындар жұмыс атқаруды күтеді. (ағында құру шығындары, қолданылған ағынды жою; белсенді ағындар санына шектеудің болмауы)

КОНКУРЕНТТІЛІК

- Тізбекті программалау – бір уақытта бір тапсырманы орындау
- Конкуренттік программалау – бір уақытта бірнеше тапсырманы орындауды білдіреді және аппараттық қолдауды қажет етпейді.
- Параллелизм - қосымшаның жұмыс жылдамдығын арттыру үшін бір мезгілде әртүрлі ядроларда бірнеше тапсырмаларды орындау тұжырымдамасы. Барлық параллель программалар конкурентті болады, бірақ конкурентті программа параллель болуы міндетті емес. Параллелизмге көп ядролы құрылғыларда ғана қол жеткізуге болады.
- Көпағындық – бірнеше атқару ағынын пайдаланатын конкуренттілік формасы.
- Асинхронды программалау – функцияның атқарылу нәтижесі лезде емес, бірқатар уақытта белгілі болатын конкуренттілік формасы.

ЖАДЫ

- Орталық процессорды (ОП) жоспарлау арқасында ОП пайдалану тиімділігін және компьютердің жауап беру жылдамдығын арттыруға болады. Алайда бұл жағдайда жадыда көптеген процестерді сақтау қажет болады.
- Орталық процессор тікелей байланыса алатын жалпы мақсаттағы қоймалар:
- Регистрлер – процессор ядросына кірістірілген
- Кэш – процессор чипына кірістіріледі
- Негізгі жады – жады шинасындағы транзакциялар арқылы қолжетімді
- Атқарылатын нұсқаулар және осы нұсқауларда қолданылатын мәліметтер тікелей қатынасу құрылғыларының бірінде орналасуы тиіс.

ЖАДЫ

- Жүйенің дұрыс жұмысы үшін операциялық жүйені қолданушылық процестердің әрекеттерінен және қолданушы процестерді бір-бірінің өзара әрекеттерінен қорғау керек. Әрбір процестің жеке жады кеңістігі болуы тиіс, яғни процесс қатынаса алатын адрестер диапазонын анықтап, процесс тек солармен жұмыс істеуін қамтамасыз ету қажет. Осы мақсатта базалық (физикалық жадының ең кіші мүмкін адресі) және шектік регистр (диапазон өлшемі) қолданылады. Процестің базалық және шектік регистрлерін бек операциялық жүйе жүктей алады.
- Программа процессорда атқарылуы үшін тікелей қатынасу жадысында орналасуы қажет. Егер олай болмаса, онда ол осы жадыға көшіріледі.
- Орталық процессор генерациялаған адрес логикалық деп аталады, ал жады блогына көрінетін адрес физикалық деп аталады. Логикалық және физикалық адрестерді байланыстыру уақытына қарай, бұл адрестер өзара сәйкес немесе әртүрлі болуы мүмкін. Жадыны басқару құрылғысы осы адрестер арасындағы өзара бейнелеуді орындайды.

ЖАДЫ

- Жадыны операциялық жүйеге және қолданушылық процестерге беру үшін, жадыны бөлшектеуге тура келеді.
- Жадыны үздіксіз бөлу – процеске қажетті жады бөлігін беру, процесс жадының үздіксіз аймағына орналасады.
- Жадыны үлестіру стратегиялары:
- *First fit – алғашқы сәйкес аймақты бөлу*
- *Best fit – барынша сәйкес аймақты бөлу*
- *Worst fit – нашар орналастыру*

ФРАГМЕНТТЕУ

- FF және BF стратегиялары сыртқы фрагменттеуді тудырады. Бос жады кеңістігі кіші бөліктерге бөлінгеннің әсерінен, бос жады аймағы жеткілікті көлемді болғанымен, үзіліссіз бос аймақтың жеткіліксіздігінен сыртқы фрагменттеу пайда болады.
- Жалпы жағдайда, процестер арасында қалған бос фрагменттер көлемі аз болса, ол көлем фрагментті бақылау шығынын өтей алмауы мүмкін.
- Бұл мәселені шешу үшін жадыны бекітілген көлемдегі блоктарға бөліп, жадыны осы блоктар өлшемдерінің негізінде үлестіру. Мұндай жағдайда блок құрамындағы бос қалған жады ішкі фрагменттеуді тудырады.
- Сыртқы фрагменттеуді шешудің мүмкін жолдарының бірі – жадыны сығу – жадының құрамын қайта орналастыру. Бірақ процестердің жадыда орналастырылуы статикалық болса, сығуды орындау мүмкін емес. Сонымен қатар, сығу схемасы қымбатқа түсуі мүмкін.
- Сыртқы фрагменттеу мәселесін шешудің басқа жолы – процестердің логикалық адрестік кеңістігі үздіксіз болмауына жол беру. Бұл стратегия пейджингте қолданылады.



Назарларыңызға рахмет!