Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

**Кэш–жады. КЭШ–тың жұмыс механизімі.**

(сабақ тақырыбы)

**Модуль/Пән атауы** Ішкі тапсырмалардың өзара әрекеттесу сызбасын жасап, оларды бір тапсырмаға әзірлеу

**Дайындаған педагог** Нургисаева У.М

**20\_**25**\_ жылғы** «\_24\_\_\_» \_\_ақпан \_\_\_\_

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 2 курс, 2БҚ-23

Сабақ түрі: Тәжірибелік

**2. Мақсаты, міндеттері:**

**Оқу:** Кэш-жадының жұмысы мен құрылымы, кэш-хит және кэш-мисс процестері, оның процессор өнімділігіне әсерін қарастыру. L1, L2 және L3 кэштердің айырмашылықтары, әрбір деңгейдің рөлі мен тиімділігін анықтау.

**Дамыту:** Логикалық ойлау қабілетін жетілдіру. Электрондық құрылғылардың жұмыс істеу принциптерін терең түсіну.

**Тәрбиелік:** Ұқыптылық пен жауапкершілікті қалыптастыру.

**3. Оқу-жаттығу процесінде білім алушылар меңгеретін күтілетін нәтижелер және кәсіби дағдылар тізбесі:** Кэш-хит пен кэш-мисс процестерін түсініп, олардың компьютер жүйесінің жұмыс жылдамдығына әсерін талдай алады. Кэш механизмдерін зерттеу арқылы әртүрлі жағдайлардағы тиімділікті бағалауды үйренеді

**4. Қажетті ресурстар:** ДК немесе ноутбуктер

**Токеймбетов Б.Т., Қасымбеков А.Б.** – Цифрлық құрылғылар және микропроцессорлар негіздері. Алматы: Қазақ университеті, 2018.

[***https://libr.aues.kz/facultet/frts/kaf\_e/2/umm/e\_3.htm***](https://libr.aues.kz/facultet/frts/kaf_e/2/umm/e_3.htm)

***5. Сабақтың барысы: (90 минут)***

**5.1. Ұйымдастыру кезеңі:** *( 3 мин )*

**5.2. Үй жұмысын жан-жақты тексеру:**

***“Миға шабуыл ” әдісі*** *(15 минут)*

**Кэш–жады. КЭШ–тың жұмыс механизімі.**

Заманауи компьютерлік жүйелердің өнімділігі көбінесе жедел жадтың (RAM) жылдамдығына және процессордың (CPU) жұмыс істеу тиімділігіне байланысты. Дегенмен, орталық процессор (ОП) мен жедел жад арасындағы деректер алмасу жылдамдығы бірдей емес, өйткені жедел жады салыстырмалы түрде баяу жұмыс істейді. Осы мәселелерді шешу үшін компьютерлік архитектурада **кэш-жады** (cache memory) қолданылады.

Кэш-жад – бұл **жоғары жылдамдықты буферлік жады**, оның негізгі мақсаты – процессор мен негізгі жедел жад арасындағы деректер алмасу жылдамдығын арттыру. Кэш-жадтың болуы компьютердің өнімділігін арттырады және процессордың күту уақытын азайтады.

Кэш-жадтың пайда болу тарихы 1960-жылдары компьютерлік архитектураның дамуымен байланысты. Бұл кезеңде процессорлар мен негізгі жедел жад (RAM) арасындағы деректер алмасу жылдамдығындағы үлкен айырмашылық айқын көріне бастады, өйткені процессорлар есептеу қуатын арттырғанымен, жад технологиялары салыстырмалы түрде баяу дамыды. Осы мәселені шешу мақсатында ғалымдар уақытша деректерді жылдам қолжетімді буферлік жадта сақтау идеясын ұсынды. Алғашқы кэш-жадтар IBM компаниясының 360/85 моделі сияқты мейнфреймдерде қолданылып, тиімділігін дәлелдеді. 1970-1980 жылдары бұл технология жетілдіріліп, суперкомпьютерлер мен миникомпьютерлерде қолданыла бастады. 1990-жылдары дербес компьютерлердің қарқынды дамуы нәтижесінде кэш-жад орталық процессорлардың ажырамас бөлігіне айналып, көпдеңгейлі (L1, L2) архитектуралар пайда болды. 2000-жылдардан бастап кэш-жадтың үшінші деңгейі (L3) енгізіліп, процессор ядроларының өнімділігін одан әрі арттыруға мүмкіндік берді. Қазіргі таңда кэш-жад жасанды интеллект, үлкен деректерді өңдеу және жоғары жылдамдықты есептеу жүйелерінде маңызды рөл атқарады.

Кэш жазбалар жиынтығынан тұрады. Әрбір жазба деректер элементімен немесе деректер блогымен (деректердің кішкене бөлігі) байланысты, бұл негізгі жадтағы деректер элементінің көшірмесі. Әрбір жазбада бар [идентификатор](https://translated.turbopages.org/proxy_u/ru-kk.ru.27cb33a5-67b95d3d-c58b3535-74722d776562/https/ru.wikipedia.org/wiki/%D0%98%D0%B4%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80), жиі деп аталады [тегпен](https://translated.turbopages.org/proxy_u/ru-kk.ru.27cb33a5-67b95d3d-c58b3535-74722d776562/https/ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%B3_(%D0%BC%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5)), бұл кэштегі мәліметтер элементтері мен олардың негізгі жадтағы көшірмелері арасындағы сәйкестікті анықтайды.

Кэш клиенті (орталық Есептеуіш Бөлім, веб-шолғыш, амалдық жүйе) деректерге қол жеткізген кезде, ең алдымен кэш зерттеледі. Егер кэште сұралған деректер элементінің идентификаторына сәйкес келетін идентификаторы бар жазба табылса, онда кэштегі деректер элементтері қолданылады. Мұндай жағдай деп аталады *кэшті соғу арқылы*. Егер кэште сұралған деректер элементі бар жазба табылмаса, онда ол негізгі жадтан кэшке оқылады және келесі қоңырауларға қол жетімді болады. Мұндай жағдай деп аталады *кэшті жіберіп алу арқылы*. Нәтиже табылған кезде кэшке кіру пайызы деп аталады *соққы деңгейімен*, немесе *соққылар коэффициентімен* кэшке.

Мысалы, веб-шолғыш дискідегі жергілікті кэшті сұралған URL мекен-жайына сәйкес келетін веб-парақтың жергілікті көшірмесін тексереді. Бұл мысалда URL мекен—жайы идентификатор, ал веб-парақтың мазмұны мәліметтер элементтері болып табылады.

Егер кэштің көлемі шектеулі болса, жіберіп алған жағдайда, орынды босату үшін кейбір жазбаларды тастау туралы шешім қабылдануы мүмкін. Тасталатын жазбаны таңдау үшін әр түрлі қолданылады [алдын ала алгоритмдер](https://translated.turbopages.org/proxy_u/ru-kk.ru.27cb33a5-67b95d3d-c58b3535-74722d776562/https/ru.wikipedia.org/wiki/%D0%9A%D1%8D%D1%88#%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%B2%D1%8B%D1%82%D0%B5%D1%81%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F).

Кэштегі мәліметтер элементтері өзгертілген кезде, олар негізгі жадта жаңартылады. Кэштегі деректерді өзгерту мен негізгі жадыны жаңарту арасындағы уақыттың кешігуі деп аталатынмен басқарылады [саясатпен жазбалар](https://translated.turbopages.org/proxy_u/ru-kk.ru.27cb33a5-67b95d3d-c58b3535-74722d776562/https/ru.wikipedia.org/wiki/%D0%9A%D1%8D%D1%88#%D0%9F%D0%BE%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B0_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D0%B8_%D0%BF%D1%80%D0%B8_%D0%BA%D1%8D%D1%88%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B8).

Кэште бірге *дереу жазумен* әрбір өзгеріс негізгі жадтағы деректердің синхронды жаңартылуын тудырады.

Кэште бірге *кейінге қалдырылған жазбамен* (немесе *кері жазбамен*) жаңарту деректер элементі ығыстырылған жағдайда, мерзімді түрде немесе клиенттің сұрауы бойынша жүзеге асырылады. Өзгертілген деректер элементтерін қадағалау үшін кэш жазбалары модификация атрибутын сақтайды (*өзгертілген* немесе *"лас"*). Кешіктірілген кэшті жіберіп алу үшін негізгі жадқа екі қол жетімділік қажет болуы мүмкін: біріншісі кэштен ауыстырылатын деректерді жазу үшін, екіншісі қажетті деректер элементін оқу үшін.

**2. Кэш-жадының анықтамасы және құрылымы**

Кэш-жады – бұл **аз көлемді, бірақ өте жылдам жад түрі**, ол процессордың ең жиі қолданылатын деректерін уақытша сақтау үшін қолданылады.

**Кэш-жадының құрылымдық компоненттері**

Кэш-жады бірнеше негізгі бөліктерден тұрады:

1. **Тегтік жад (Tag Memory)** – кэшке сақталған деректердің негізгі жадтағы мекенжайын сақтайды.
2. **Деректер жад (Data Memory)** – нақты сақталатын деректер орналасқан аймақ.
3. **Басқару логикасы (Cache Controller)** – процессор мен кэш-жадының байланысын басқаратын модуль.

**Кэш-жадының деңгейлері**

Кэш-жады бірнеше деңгейден тұруы мүмкін:

* **L1 (Level 1) – Бірінші деңгейлі кэш**
  + Ең жылдам, бірақ ең аз көлемді (әдетте 16-64 КБ).
  + Тікелей процессор өзегіне (core) интеграцияланған.
* **L2 (Level 2) – Екінші деңгейлі кэш**
  + L1-ге қарағанда баяу, бірақ көлемі үлкенірек (128 КБ – 1 МБ).
  + Кейбір процессорларда әр өзекке бөлек, кейде ортақ қолданылады.
* **L3 (Level 3) – Үшінші деңгейлі кэш**
  + Барлық процессор өзектеріне ортақ болады.
  + L1 және L2-ге қарағанда баяу, бірақ көлемі үлкен (4-64 МБ).
* **L4 (Level 4) – Төртінші деңгейлі кэш (сирек кездеседі)**
  + Жүйелік жады контроллерімен байланыстырылады.

Кэш деңгейлері процессордың жұмысын жеделдетіп, деректерді негізгі жадтан алуға кететін уақытты азайтады.

**3. Кэш-жадының жұмыс механизмі**

Кэш-жады процессордың жұмыс жылдамдығын арттыру үшін белгілі бір алгоритмдер бойынша жұмыс істейді.

**3.1. Кэш-жадтағы деректерді іздеу процесі**

1. **Процессор деректерді сұрайды.**
2. **Кэш-жады алдымен іздеу жүргізеді (cache lookup).**
   * Егер қажетті деректер кэште болса (**cache hit**) – процессор оларды бірден алады.
   * Егер кэште жоқ болса (**cache miss**) – деректер негізгі жедел жадтан алынып, кэшке жазылады.
3. **Деректер өңделгеннен кейін, олар қайтадан кэшке немесе негізгі жадқа сақталады.**

**3.2. Кэш-жадының тиімділігін анықтайтын факторлар**

* **Кэштің соққы жылдамдығы (cache hit rate):** Кэштен табылған деректердің жалпы сұраныстарға пайыздық қатынасы.
* **Кэштің қателік жылдамдығы (cache miss rate):** Кэштен табылмаған деректердің жалпы сұраныстарға пайыздық қатынасы.
* **Деректерді алмастыру стратегиялары (cache replacement policies):** Кэш толған кезде, қай деректерді жою керектігін анықтайтын алгоритмдер.

**4. Кэш-жадының басқару стратегиялары**

Кэш-жады тиімді жұмыс істеуі үшін арнайы алгоритмдер қолданылады.

**4.1. Кэш-орналасу стратегиялары (Cache Mapping Techniques)**

Кэшке қандай деректер жазылатынын және олардың қайда сақталатынын анықтайтын әдістер:

* **Тікелей байланыстырылған кэш (Direct Mapped Cache)**
  + Әрбір жады мекенжайы кэштің нақты ұяшығына жазылады.
  + Жүзеге асыру оңай, бірақ **кэш қақтығыстары (cache collision)** көп болады.
* **Ассоциативті кэш (Fully Associative Cache)**
  + Деректер кез келген бос ұяшыққа жазылады.
  + Икемді, бірақ күрделі және қымбат.
* **Қисынды ассоциативті кэш (Set-Associative Cache)**
  + Тікелей байланыстырылған және толық ассоциативті әдістердің гибриді.
  + Көптеген процессорлар осы әдісті қолданады.

**4.2. Кэшті алмастыру стратегиялары (Cache Replacement Policies)**

Кэш толған кезде қай деректерді жою керектігін анықтайтын алгоритмдер:

* **LRU (Least Recently Used)** – Ең сирек қолданылған деректер жойылады.
* **FIFO (First-In, First-Out)** – Бірінші келген деректер бірінші жойылады.
* **LFU (Least Frequently Used)** – Ең аз қолданылған деректер жойылады.
* **Random Replacement** – Деректер кездейсоқ жойылады.

**5. Кэш-жадының артықшылықтары мен кемшіліктері**

**Артықшылықтары:**

✅ Процессордың деректерге қол жеткізу уақытын қысқартады.  
✅ Компьютердің жалпы өнімділігін арттырады.  
✅ Жиі қолданылатын деректерді жылдам қайта алу мүмкіндігін қамтамасыз етеді.

**Кемшіліктері:**

❌ Кэш-жадтың сыйымдылығы аз.  
❌ Аппараттық іске асыру күрделі және қымбат.  
❌ Қате стратегиялар қолданылса, кэштің тиімділігі төмендейді.

Кэш-жадыны қолдану қажеттілігі процессор мен компьютер жадының әртүрлі бөлімдері арасындағы ақпарат алмасу жылдамдығының айырмашылығымен түсіндіріледі. Кез-келген қосымшаның жұмысы қажетті деректерді салыстырмалы түрде баяу қатты дискіден жедел жадқа (компьютердің жедел жадына) динамикалық кездейсоқ қатынас бөліміне жіберуден басталады. Сол жерден оларды процессордың микросхемасында орналасқан L2 кэшіне (L2 жады) немесе процессордың жанында орналасқан жоғары жылдамдықты бөлек SRAM чипіне беруге болады. Сонымен, ең көп қолданылатын ақпаратты процессордың арнайы бөлімі болып табылатын L1 кэшіне (бірінші деңгейлі жадқа) беруге болады. Бірінші деңгей кэшінің мөлшері тек 128 КБ шамасында, ал екінші деңгей қазірдің өзінде 512 КБ құрайды. Салыстыру үшін оперативті жадының өлшемі 1 ГБ болуы мүмкін. Кез келген команданың орындалуы белгілі бір схема бойынша жүреді: - ақпараттың регистрлерін талдау; - бірінші деңгейдегі кэштің мәліметтерін сканерлеу; - кэш туралы ақпаратты тексеру екінші деңгей; - негізгі жадының деректерін талдау; - қатты дискінің жадына қол жеткізу. Процессордың қажетті деректерді алуға кеткен уақыты ақпарат сақталатын жерге тікелей пропорционалды. Осылайша, бірінші деңгейдегі кэшке қол жеткізу 1-ден 3 циклға дейін, екінші деңгей - алтыдан он екі циклге дейін, ал негізгі жадқа - ондаған, ал кейбір жағдайларда - жүздеген циклды алады. Кэш жады сервердің жұмыс істеу процесінде ерекше рөл атқарады, өйткені жадқа трафиктің трафигі бұл жағдайда маңызды болуы мүмкін. Кэш құрылымы сонымен қатар жыл сайын 50 пайызға өсетін процессор жылдамдығы мен тек 5 пайызға өсетін жедел жады деректері арасындағы айырмашылықты азайту үшін қызмет етеді. Кэш жадының үшінші және төртінші деңгейлерінің үздіксіз дамуы осы бағыттағы қисынды қадамдар сияқты. Дамудың тағы бір мүмкін бағыты кэш-жадыны бағдарламалық басқаруға көшу болуы мүмкін.

**Тапсырмалар**

1. **Кэш-жады туралы теория**:
   * Кэш-жады — процессор мен негізгі жады арасында орналасқан жылдам жады түрі.
   * Кэш-жадының негізгі мақсаты — процессорға жиі қолданылатын деректерді тез жеткізу.
   * Әдетте кэш-жадының бірнеше деңгейі болады: L1, L2, L3. L1 — ең жылдам, бірақ кішкентай, ал L3 — үлкенірек, бірақ баяулау.
   * Кэш-жадының жұмыс принципі: кэш-хит (дерек кэште бар) және кэш-мисс (дерек кэште жоқ).
2. **Тапсырма**:
   * Кэш-хит және кэш-мисс жағдайларында деректердің қалай өңделетінін түсіну.
   * Кэшті Python тілінде модельдеп, кэш-хит пен кэш-мисс жиілігін есептеу.
   * Кэш-жадының тиімділігін зерттеу және кэштің әсерін түрлі стратегиялармен (мысалы, FIFO, LRU) салыстыру.
3. **Алгоритмдер**:
   * **FIFO (First In, First Out)**
   * **LRU (Least Recently Used)**

**Модельдеу және код:**

Python тілінде кэш-жадының жұмысын модельдеп, кэш-хит және кэш-мисс жиілігін есептейтін бағдарлама жасау.

|  |  |  |
| --- | --- | --- |
| **Білдім** | **Білемін** | **Білгім келеді** |
|  |  |  |

**Рефлексия**

**Үй тапсырмасы –** Практикалық жұмыс №3 орындау және қорғау