

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут»

Факультет прикладної математики

Кафедра спеціалізованих комп'ютерних систем

КОНСПЕКТ ЛЕКЦІЙ
з дисципліни

**ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ
ОБЧИСЛЕННЯ**

Лектор: доцент Марченко О.І.

Copyright © 2009 – 2014, Марченко О.І.

Київ, 2009–2014

Класифікація РНІО

Одну з перших масив. в КС побудував у 1966 р. співробітник фірми IBM на той час лідаже Райн.

Це, як класифікація базується на сумі потоку даних

над. в КС, що поділяється на:

1) поток даних які пассять та
працюють (-ами) та

2) поток команд, які виконують процесор (-и)
при увличному виконанні з них потоки даних та
команд можуть бути однією з двох множин
також будуть залежні від процесора. Відповідно до КС
поміж ними відрізняють:

1 ОКДА (SISD) - КС з одноточковим потоком
команд і даних.

2. МКД (MISD) - КС з множинним потоком
команд і одноточковим потоком даних

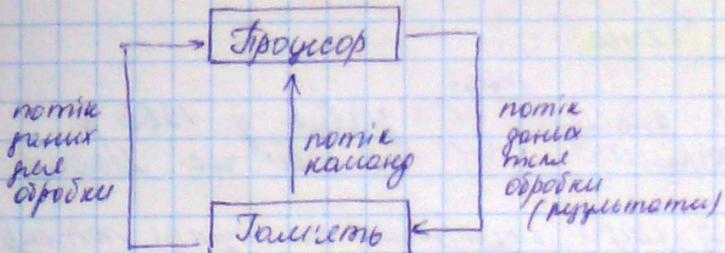
3. ОКМД (SIMD) - КС з одноточковим потоком даних
команд і множинним потоком даних

4. МКМД (MIMD) - КС з множинними потоками
команд і даних

Поток команд - послідовність команд, які виконуються
КС.

Поток даних - послідовність даних, що з них
звертається потоком команд, викинутою з
працюючими результатами.

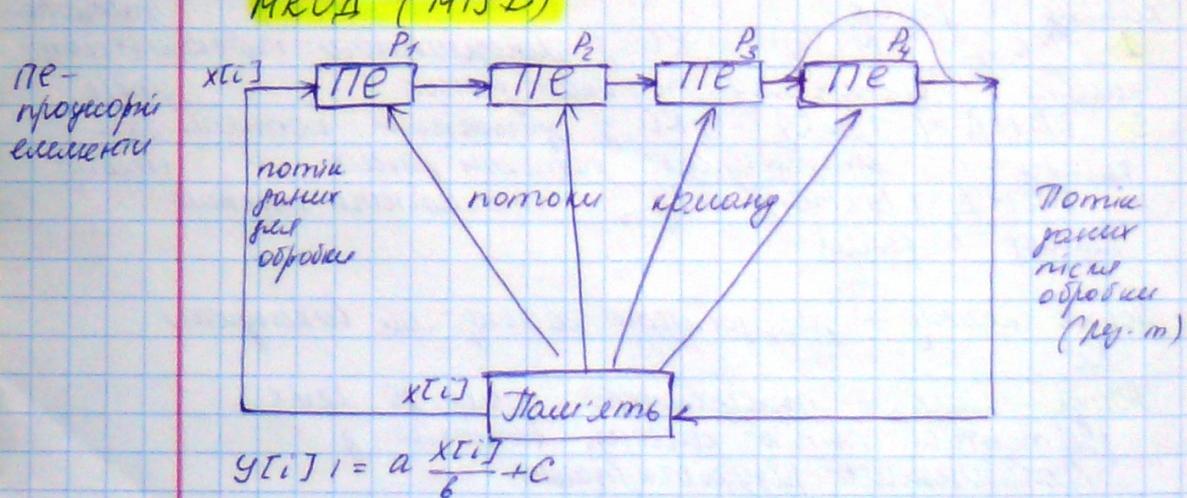
ОКОД (SISD)



до класу систем ОКОД належать також паралельно-послідовні комп'ютери з архітектурою Ронана фан Нейланда, але варіантно - а їх не називають ПК.

Паралельні програми можуть бути виконані також і на системах типу ОКОД, але в умові будьодно маємо спаду не з мобільної паралельності, а (чевічше) півдроздільності.

МКОД (MISD)



$$P_1 := a x[i]$$

$$P_2 := P_1 / b$$

$$P_3 := P_2 / b + c$$

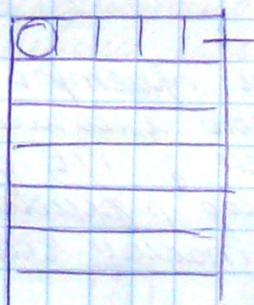
В системі МКОД процесор обробляє подільовані не залежні етапи, концесію з яких використовує окремий процесорчики підсистеми PE складають сукупність магістралей обробки (конвеєр процесорів) з етапі.

результатами, обробленими і опущеними.

Розділ 5 обробки даних та використання
побудованої магістральної обчислювальної
системи високої розподіленості при виконанні
різних лінійних програм з одновід-
повідними операціями. При частих перериваннях
або зупинках комп'ютера розподілені
швидкість обробки даних знижується.

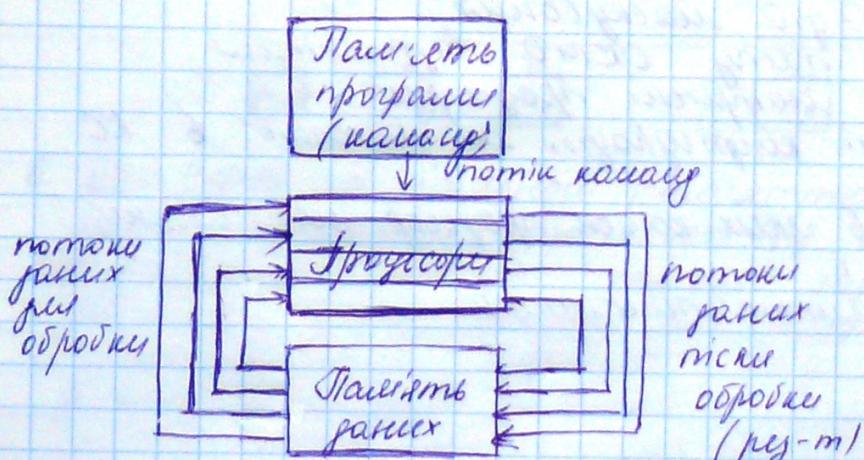
"Часті" системи типу МКД (тобто в
їхніх обсягом відсутні вик. на магістраль
процесорів) існують в основному як експерим.
проекти, але конвеєрний принцип обчислень
часто використов. як складова частина / опорний
блок в складних обчисл. системах.

Принцип конвеєризації

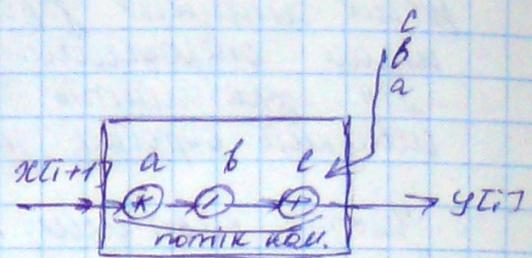
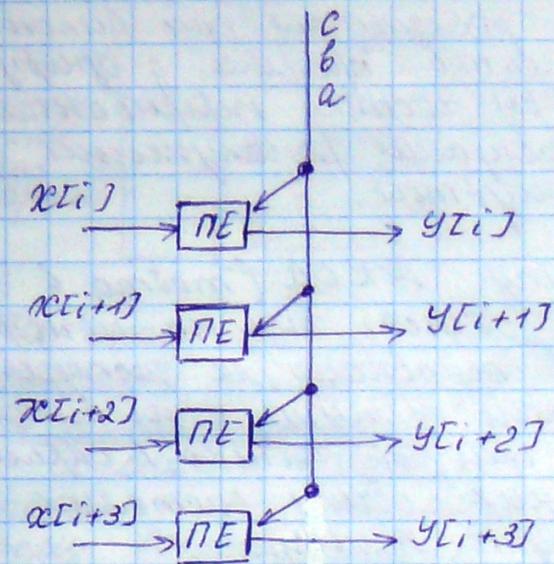


4
вибірка пам'яті
змінн
пам'яті
завантаження даних 2
виконання пам'яті 1

ОКМД (SIMD)



$$y[i] = \frac{a x[i]}{b} + c$$



В системах типу ОКМО один пристрій керується виконує керув. роботою матричних процесорів та та, що поширені з PE однієї з тих послідовностей калькулятора над результатами попередніх, виконуючи такий чин чиєрочний обчисл. порядк. процес.

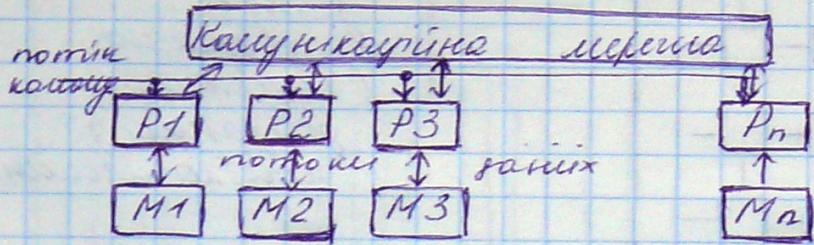
Тут же робота всіх PE може бути синхронізована неподрібність всіх калькуляторів, що виконують деякий оп-такт залежності.

До систем типу ОКМО відносяться висторії та матричні процесори.

Е єдні головні конфігурації, які джестосов. є КС ОКМО:

- 1) системи, що всіх калькуляторів мат. власну лок. пам'ятю і
- 2) КС є спільнотою пам'яті то.

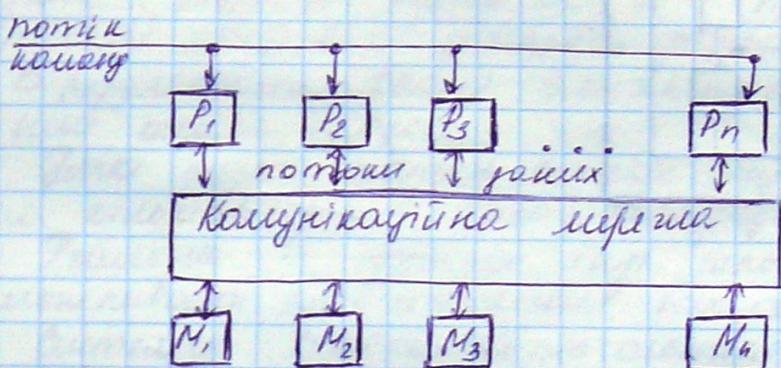
КС з ОКМО з лок. пам'ятю для кількох процесорів.



Процесори буджетного пам'ятої з'єднуються одним з одиниць зчіпу **каскадна пам'ять**. Якщо усі процесори чияють з'єднувати з їхнім заданим пам'яті процесором, то може обсягнеться зчіп працюючими процесорами.

Використано архіт. б ІІІТАС IV (один з перших базових процесорних касп.)

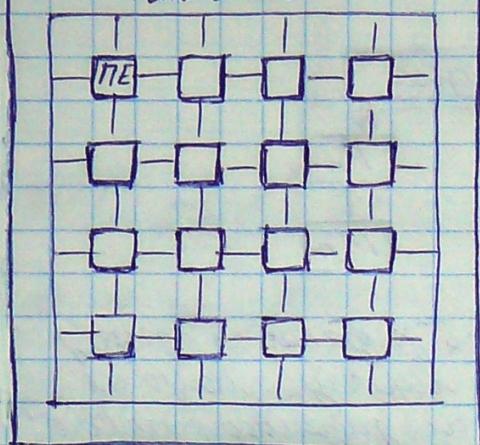
КС ОКМО з спільнотою пам'ятю



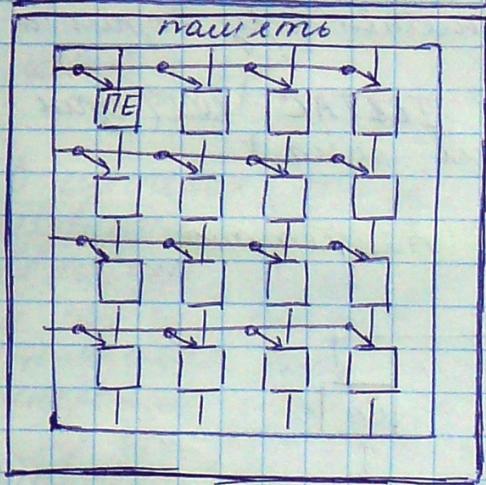
В КС ОКМО з спільнотою пам'ятю процесори можуть з'єднуватися самі собою без зчіп. пам'яті. Два процесори можуть передавати якісні одиниці даних зчіп процесором до іншого пам'яті. Це спрощення вартоїстю існує архітектура з одиницею блоком пам'яті. Вторий процесор був називаний (-x)

Матричні процесори
паштет

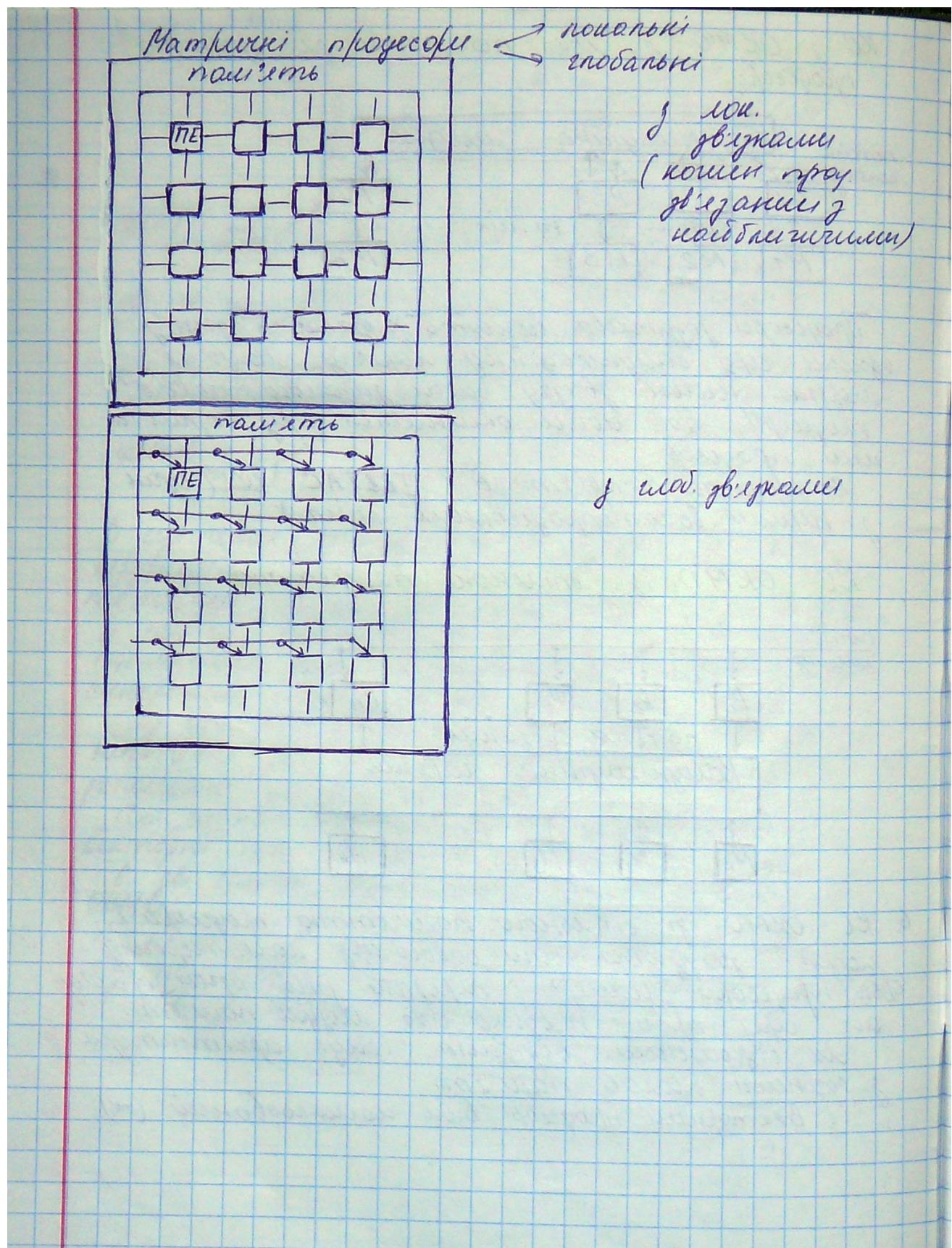
помаранчі
глобальні



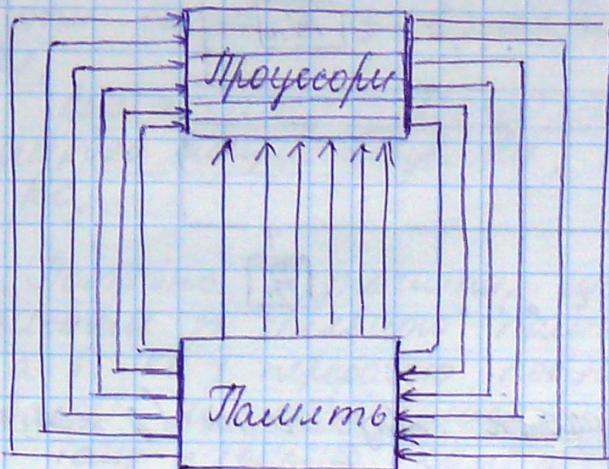
↓ мон.
зв'язки
(кошик з роз-
з'єднанням з
насібничими)



↓ мон. зв'язки



МКМД (MIMO)



В ум. сист. можна прац. із стійкими потоками даних; потоками різних. В таких системах звичайно пристріб кер. виконують узварнісіс одноголосими високочастотними різних фреквеменів (одині інші не працюють (ні прац. або потоків) або всіх. звичайнох різних програм). В результаті реаліз. асинхронній паралельні обчисл. процес.

Задачі сист. мають велику швидкість та вис. кількість різнов. сист. планують.

Реальнісіс в обробці інф. започаткову вис. маневривості завантаженням всіх процесорів одноголоско.

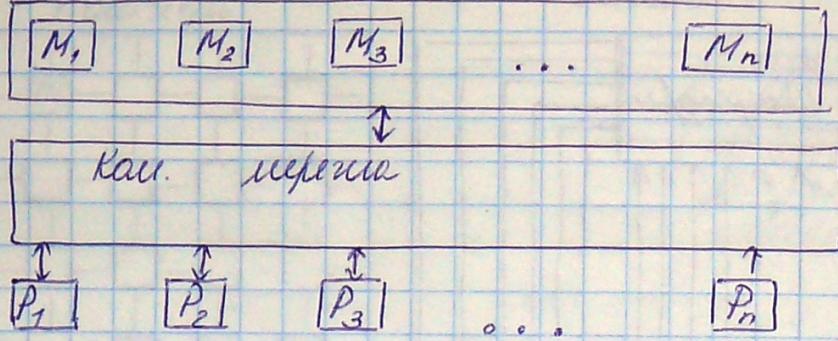
Системи МКМД виконують високою маневривістю високі рівні маневривості високого резервування організаціїх процесорів та пристріб.

КС МКМД складається з багатьох процесорів та багатьох блоків пам'яті, з'єднаних до головного кашукінгаїсії лінії.

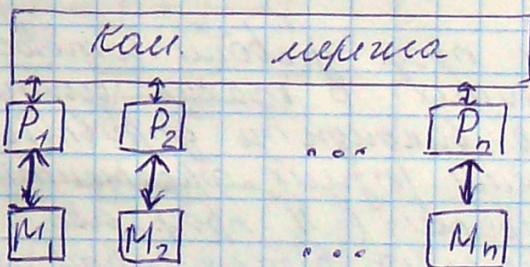
КС МКМД поділено на 2 :

- 1) зі спільнокою пам'яттю
- 2) з незалежною пам'яттю.

КС МКМД є спільного пам'ятю.



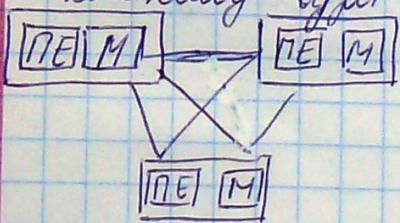
КС МКМД з паралельною побудовою (з кон. пам'ятю)



В першій групі архітектур процесори обсл. інформ. зерку спільної пам'яті, при чому

Інакі архітектури надають синхронізацію базамопроцесорними системами.

Касп. Сист. 2-ї групи обсл. інформ. зерку кашук. мережею КС є паралельною побудовою системи з розподіленого пам'ятю, за якої маємо кон. пам'ятю і оптимізовані процесори у консистентності вузлів кашук. мережі.



В такій сист. виробляється спільна пам'ятю, тому перенесувати її з одного до іншої кон. пам'яті потрібно за допом. побудованою.

Механізми передачі побічного розміщення складають послання-відповідь опу. касину що побудовані бути будовани у вигляді проходів забезпечено.

Архітектуру з розподільчою пам'яткою побудувавши шевченко використовувати, коли її винесе великий КС.

Помірно відзначити, що програмування системи є спільною пам'яткою в простій сенсі, а в КС є передача побудованої країні розміщені місця відповідності.

Таку з'явилася ходінкова КС.

У системах передачі побудованої країні відповідь може бути за сок. касини та вже базайонроесорної системи так і зорногількою розподіленої кашинометрикою системою.

16.02.10

Однією з найкорисніших технологій розподільчих обчислень є **grid-computing** (grid-computing)

grid-мережа

Концепція. Grid буде створено на таких основних пам'ятках:

1) широке і постійне розширення продуктивності процесорів для кашинометрів масового виробництва.

Широкої спасової кашинометрії = широкості однієї спеції супер-кашинометра 10р. давності

2) наявні широких оптимальних зважок. Сучасні барви чиї зважки зберуть країні

Систематичні масові пропускни розширення 10Гб/сек і підтримка високих. А низко-частотні кашинометрів різних організацій ю країні відбудеться на широкості 1-2 Гб/сек

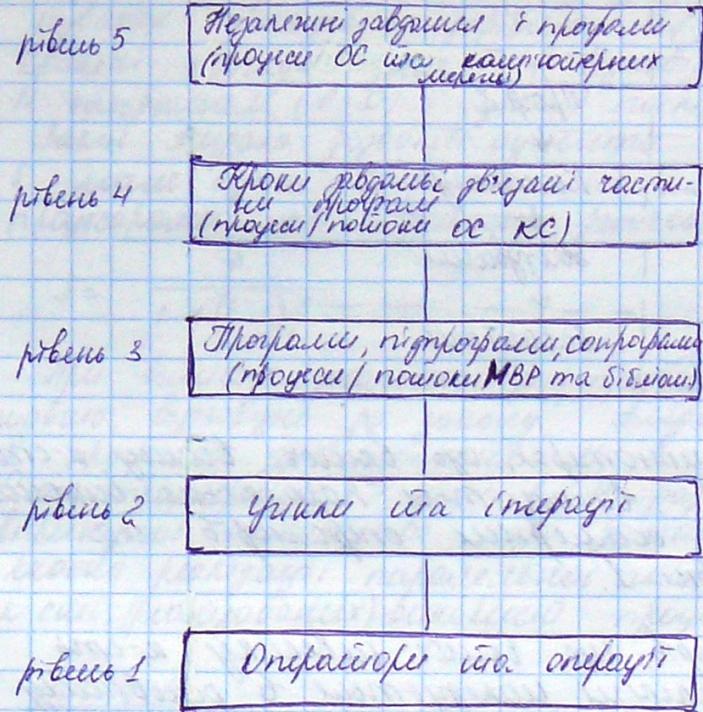
3) розширені інтернету, а також глобальні проекти обміну використаного та заміненої світової економіки.

4) постійне впровадження технології та яскраві дії. Альтернативи.

По-сумі Тріг є "надбудовою над інтерпретацією"
Основне представлення Трігу - організує розподілення
обчислень для розв'язання складних задач наук і
техніки, які вимагають великих обчисл. ресурсів,
а саме підчленені, ресурсів розрізані, таєм
обчислень.

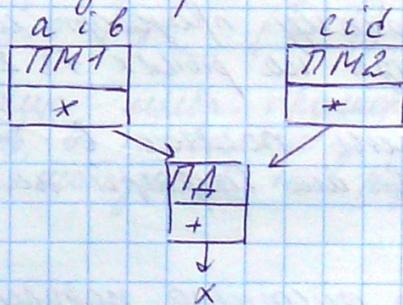
Навірною єї структурної пам'ятки
чиєю рисункою Тріг є сироговно-редкованою
чубко хачуго, Тріг можна називати однією з
найінтересніших.

Рівні розмежування



MBP -
motor
excocore
fibres

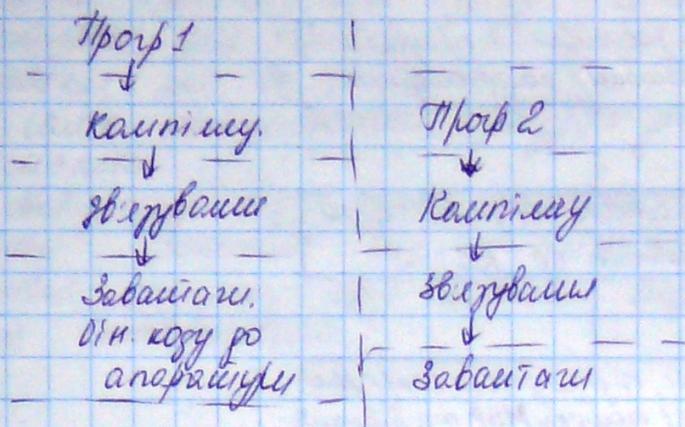
Присоединение к правой части: $x := ax + cx$



Приклад з рівніє
(приклади з величорівих та малюнків
однаково)

Приклад 3 робота (інженерного консультанта) з
макетами

лічани програми;



В багатьох комп'ютерах, що вміють виконувати паралельні процесорів, виконуються паралельні виконання низких послідовних операцій та операцій на рівні 1.

Деякі комп'ютери, що вміють виконувати паралельні процесорів, паралелізм реалізується в основному за допомогою обміну інф. між процесорами на високих рівнях 3, 4, 5.

Крім цього, в комп'ютерах може бути реалізований паралелізм на рівнях 1 і 2.

Високорівнева обробка отриманої пасивної вимоги в межах однієї послідовності дій може бути реалізована на рівні 2.

В комп'ютерах з високим асистентом паралелізму II обробка вим. головного циклу може бути реалізована на низких рівнях 1, 2 (наприклад, підтримкою процесорів)

Закон Аїрулі є другою послідовністю
У реальній програмі зазвичай операції, які передається вико-
нувачем постійно, зорваних f , $0 \leq f \leq 1$.
Кращі випадки та $f = 0$ і відповідность умови
 $f = 0$ і умови постійності ($f = 1$).
Закон Аїрулі дозволяє отримати все прискорення
 s може бути обчислена за наступнієї з р.
підсумуваннями при додаванні значені f .

$$s = \frac{r}{1 + (r-1)f}$$

При великих значеннях r , прискорення $s \rightarrow \frac{1}{f}$,
тобто, відповідно до закону Аїрулі, якщо $\frac{9}{10}$ програми здійснюються f , а $\frac{1}{10}$ постійно,
що прискорення \geq ніж в 10 разів одержані
виконані післякої по жа здійснені від
еконії реалізації паралельної частини коду;
к-сії (реалізуваних) використані процесори.

Наслідок:

Все цього, щоб прискорити виконання програми
в 2 рази необхідно використати не менше ніж
у 2 рази $(1 - \frac{1}{2})$ -ту частину програми.

Якщо переда прискоренням програму в 100
разів, в порівнянні з її постійними варіантами,
що недобудую в 100 разів прискорити 99% коду.

Багатопроцесна / багатопотокова обробка інформації

Це покаже, що високої ступені паралелізму можна досягти при використанні КС з великою кількістю процесорів, але че дуже дорого, тимут лише складних, кількох масивів розподіленої обробки, які мають розподілену пам'ять.

1) багатопроцесорність чи інші криєтати.

(ява процес. ядра з трьома розсіченою по тилу кристалі з використанням отворів / розподілених КЕМ-пам'яті)

2) багатопроцесна / багатопотокова обробка з хвилюванням часу (вик. кількох процесорів / потоків чи кількох процесорів.)

Процесор пам'яті може працювати з процесорами / потоками чи з хвилюванням діїковими часу (шакти)

За таке підключення виникає еф. процесору операційної системи / планувальників (шедулерів)

3) багатопроцесна / багатопотокова обробка з переключенням за поглядом.

(Переключенням будувованим при наявності привалентів пам'яті у вигляді процесорів / потоків) якщо якийсь процесор не розвивається залежно до швидкої кеш-пам'яті з більшою побічною опосередкованістю, то його призначено будувати, використовуючи такі самі ресурси процесора як інших процесорів - планувальники.

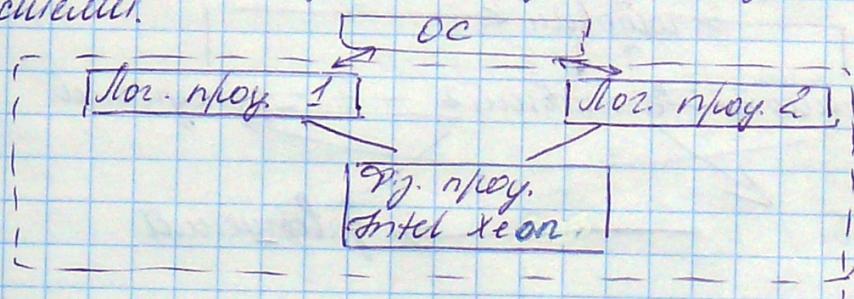
4) Однотаска багатопроцесна / багатопотокова обробка.

При такому підході процесор / потоки вик. на основу процесорів, тільки однотаска, тобто бу переключенням за кількох. Ресурси процесора розподіленої чи процесорами / потоками розподіленої, тимут чи використати, вигадані системи.

Одногасна багатопроцесорна архітектура
обробки побудована в інтерполяторну обробку
квантізації Intel. Цю технологію реалізовано в
Intel Xeon.

Технологія зручною є тим, що одногасно
швидкої частини ресурсів процесора використовують
щоб не опрацювати конкретного програмного
коду. Не використовуючи в цей час ресурси
можна роботи змінити функції, використовує
заданих за "виконанням" цього процесу/
послуги. Для цього в одину пам'ятку процесорі
можуть бути зберегти, що певного коду
своєї однокомандової ресурсів одногасного діючого
процесора.

Оскільки принципи програми "Багатій" сане 2
ел. процесори, які спрощені до підсистеми робочий
шак сано, то, у випадку побудування звичайної
системи.



23.02.10

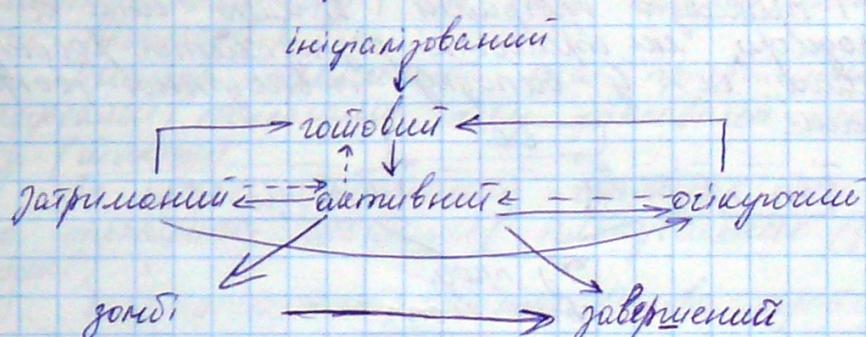
Паралельні

ВОС будають з невимісного та вимісного
багатозадачистю.

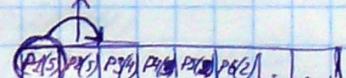
З невимісного - якщо процес джоків їде
процесора, то може інший процес не мати чого
вимісити доки він сам не буде виконавське (що
щупає отримання на екіпаж сигнал)

Вимісного - відмінні пріоритети можуть
вимісити менш пріоритетні.

Стани процесу / потоків



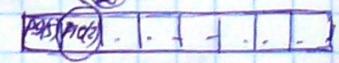
Робочим стає лише готових процесів



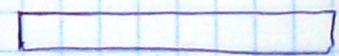
закінчених



затриманих



активних діячих



Камукалайє / синхронізація процесів / потоків:

- 1) однотаскний доступ до спільних ресурсів
- 2) несинхронна робота

чи тільки стикає час дії ресурсів, оскільки процеси перехоплюють у зоні (у надрізу з відповідними приорітетами)

Хоча зробив високий пріоритет паралельних процесів, а самога робить що буде-сь програму можна списати без "пушків".