

Завдання

Текст завдання: На запуск очікують п'ять задач. Передбачуваний час виконання задач складає 9, 6, 3, 5 і X одиниць часу. У якому порядку їх варто запустити, щоб мінімізувати оборотний час? Обчисліть оборотний час для задачі, що буде виконуватися останньою.

Розв'язок: Для мінімізації оборотного часу слід використовувати алгоритм планування „найкоротша задача – перша”, тобто:

при $X \leq 3$ порядок: $X, 3, 5, 6, 9$; Тоб $= X + 3 + 5 + 6 + 9 = X + 23$;

при $3 < X \leq 5$ порядок: $3, X, 5, 6, 9$; Тоб $= 3 + X + 5 + 6 + 9 = X + 23$;

Відповідь : „найкоротша задача – перша”, $X+23$

Завдання

Текст завдання: Порівняйте швидкість зчитування файлу через однопотоковий та багатопотоковий файлові сервери. Одержання запиту, його диспетчеризація й обробка займають 15 мс при наявності необхідних даних у блокувальному кеші. У кожному третьому випадку потрібно звертання до диска, що займає 75 мс, протягом яких потік знаходиться в стані чекання. Скільки запитів у секунду обробить а) однопотоковий сервер, б) багатопотоковий?

Розв'язок: У випадку однопоточного серверу обробка 3 запитів займає $15+15+15+75 = 120$ мс.

У випадку багатопоточного серверу обробка будь-якого запиту займає 15 мс, тому що при блокуванні потоку буде ініційовано наступний потік.

Тоді

$$N_1 = 1000 \text{ мс} \cdot 3 / 120 \text{ мс} = 25 \text{ запитів}$$

$$N_n = 1000 \text{ мс} / 15 \text{ мс} = 66 \text{ запитів}$$

Відповідь : а) 25; б) 66;

Завдання

Текст завдання: П'ять пакетних задач, А, В, С, D, Е надходять у систему практично одночасно. Очікується, що час їхнього виконання складе 10, 6; 2, 4 і 8 хв., відповідно. Їхні встановлені пріоритети складають 3, 5, 2, 1 і 4, причому 5 -- вищий пріоритет. Визначте середній оборотний час для кожного з наступних алгоритмів планування, зневажаючи часом, що затрачується на переключення між процесами:

а) пріоритетне планування;

б) "першим прийшов - першим обслугований" (у порядку 10, 6, 2, 4, 8);

в) "найкоротша задача -- перша".

Передбачається, що в кожен момент часу запущена одна задача, що працює аж до завершення. Усі задачі обмежені тільки можливостями процесора.

Розв'язок

а)

Визначимо оборотний час для кожної задачі

$$t_B = 6;$$

$$t_E = 6 + 8 = 14;$$

$$t_A = 6 + 8 + 10 = 24;$$

$$t_C = 6 + 8 + 10 + 2 = 26;$$

$$t_D = 6 + 8 + 10 + 2 + 4 = 30;$$

Та середній оборотний час:

$$t_{cp} = (6 + 14 + 24 + 26 + 30)/5 = 20.$$

б)

Визначимо оборотний час для кожної задачі

$$t_A = 10;$$

$$t_B = 10 + 6 = 16;$$

$$t_C = 10 + 6 + 2 = 18;$$

$$t_D = 10 + 6 + 2 + 4 = 22;$$

$$t_E = 10 + 6 + 2 + 4 + 8 = 30;$$

Та середній оборотний час:

$$t_{cp} = (10 + 16 + 18 + 22 + 30)/5 = 19,2.$$

в)

Визначимо оборотний час для кожної задачі

$$t_C = 2;$$

$$t_D = 2 + 4 = 6;$$

$$t_B = 2 + 4 + 6 = 12;$$

$$t_E = 2 + 4 + 6 + 8 = 20;$$

$$t_A = 2 + 4 + 6 + 8 + 10 = 30;$$

Та середній оборотний час:

$$t_{cp} = (2 + 6 + 12 + 20 + 30)/5 = 14.$$

Відповідь : а) 20; б) 19,2; в) 14;

Завдання

Текст завдання: Для прогнозування часу виконання використовується алгоритм старіння з $\alpha = 1/2$. Попередні чотири значення часу складали 40, 20, 40 і 15 мс (перше значення -- найдавніє). Зробіть оцінку наступного часу виконання

Розв'язок

$$T_{i+2} = \alpha T_i + (1-\alpha)T_{i+1}.$$

Ми маємо значення T_0, T_1, T_2, T_3 , тобто для розрахунку T_4 :

$$T_2 = \alpha T_0 + (1-\alpha)T_1,$$

$$T_3 = \alpha^2 T_0 + (1-\alpha)\alpha T_1 + (1-\alpha)T_2;$$

$$T_4 = a^3 T_0 + (1-a)a^2 T_1 + (1-a)a T_2 + (1-a) T_3 = \frac{1}{8} * 40 + \frac{1}{8} * 20 + \frac{1}{4} * 40 + \frac{1}{2} * 15 = 25 \text{ мс.}$$

Відповідь : 25 мс

Завдання

Текст завдання: У гнучку систему реального часу надходить чотири періодичних сигнали з періодами 50, 100, 200 і 250 мс. На обробку кожного сигналу потрібно 35, 20, 10 і x мс часу процесора. Вкажіть максимальне значення x, при якому система залишається такою, що піддається плануванню

Розв'язок

Умова планування системи

$$\sum_{i=1}^m \frac{C_i}{P_i} \leq 1,$$

де C_i – час обробки,

P_i – період надходження.

Тобто:

$$35/50 + 20/100 + 10/200 + x/250 \leq 1;$$

$$0,7 + 0,2 + 0,05 + x/250 \leq 1;$$

$$x/250 \leq 0,05;$$

$$x \leq 12,5 \text{ мс.}$$

Відповідь : 12,5 мс

Завдання

Текст завдання: Якщо використовується алгоритм заміщення сторінок FIFO у системі з чотирма сторінковими блоками і вісьма сторінками, скільки сторінкових переривань відбудеться для послідовності звертань 0172327103 за умови, що чотири сторінкових блоки спочатку порожні?

Розв'язок

0	1	7	2	3	2	7	1	0	3
0	1	7	2	3	3	3	3	0	0
	0	1	7	2	2	2	2	3	3
		0	1	7	7	7	7	2	2
			0	1	1	1	1	7	7
				0	0	0	0	1	1
п	п	п	п	п				п	

Відповідь : 6 переривань

Завдання

Текст завдання: Машина підтримує 48-розрядні віртуальні адреси і 32-розрядні фізичні адреси. Розмір сторінки дорівнює 8 Кбайт. Скільки потрібно записів у таблиці сторінок?

Розв'язок

$2^{48} = 256\text{T} / 8\text{ К} = 32\text{Г}$ або 2^{35} віртуальних сторінок – тобто записів у таблиці сторінок.

Відповідь : 2^{35} записів

Завдання

Текст завдання: У системі є чотири процеси і п'ять ресурсів, які можна надати процесам. Поточний розподіл ресурсів і максимальна їхня кількість, необхідна процесам, наведені у таблиці:

	Надано	Максимум
Процес А	10211	11212
Процес В	20110	22210
Процес С	11010	21310
Процес D	11110	11221

Доступно: 00x11

Яке найменше значення величини x , при якому цей стан є безпечним?
У якому порядку процеси зможуть завершити роботу?

Розв'язок

Розрахуємо «потреби» процесів:

	Потрібно
Процес А	01001
Процес В	02100
Процес С	10300
Процес D	00111

Процес D (і тільки він) може завершити роботу при $x = 1$. Тоді значення вектору доступних ресурсів зміниться на
 $00111 + 11110 = 11221$

Далі може завершити роботу процес А і вектор прийме значення $11221 + 10211 = 21432$.
Далі може завершити роботу процес С і вектор прийме значення $21432 + 11010 = 32442$.
Далі може завершити роботу процес В.

Порядок завершення процесів: D, А, С, В.

Відповідь : X=1; D, A, C, B.

Завдання

Текст завдання: Скільки часу займе завантаження з диска програми розміром 64 Кбайт, якщо його середній час пошуку дорівнює 10 мс, час обертання - 10 мс, кожна доріжка містить 32 Кбайт

а) для розміру сторінки 2 Кбайт?

б) для розміру сторінки 4 Кбайт?

Сторінки на диску розташовані випадковим образом, і кількість циліндрів є настільки великою, що можна ігнорувати варіант, при якому дві сторінки опиняться на тому самому циліндрі.

Розв'язок

32 Кбайт зчитуються за 10 мс, тобто час зчитування однієї сторінки розміром

2 Кбайт складає $2 \cdot 10 / 32 = 0,625$ мс

та 4 Кбайт -- $4 \cdot 10 / 32 = 1,25$ мс.

Загальний час завантаження підраховується як:

$T = N \text{ сторінок} \cdot (T \text{ пошуку} + T \text{ зчитування})$.

Тоді

а) $T = 64 \text{ Кбайт} / 2 \text{ Кбайт} \cdot (10 + 0,625) = 32 \cdot 10,625 = 340 \text{ мс}$;

б) $T = 64 \text{ Кбайт} / 4 \text{ Кбайт} \cdot (10 + 1,25) = 16 \cdot 11,25 = 180 \text{ мс}$;

Відповідь : а) 340 мс; б) 180 мс.

Завдання

Текст завдання: Порівняйте кількість місця, необхідного для обліку вільної пам'яті за допомогою бітового масиву і за допомогою зв'язного списку. Пам'ять розміром 128 Мбайт надається блоками по n біт (розрядів). Для зв'язного списку передбачається, що пам'ять складається з послідовності сегментів і вільних областей (кожна по 64 Кбайт), що чергуються. Будемо вважати, що для кожного вузла в зв'язному списку необхідна 32-розрядна адреса в пам'яті, 16 розрядів для довжини і 16 розрядів для поля посилення на наступний вузол. Скільки буде потрібно байтів для збереження структур у кожному випадку? При якому розмірі блоків в пам'яті використання бітового масиву буде більш економічним порівняно зі зв'язним списком?

Розв'язок

Розмір вузла зв'язного списку = 4 байта + 2 байта + 2 байта = 8 байт.

Кількість вузлів: $128\text{М} / 64\text{К} = 2\text{К}$

Розмір зв'язного списку = $2\text{К} \cdot 8 \text{ байт} = 16 \text{ Кбайт}$.

Розмір бітового масиву = $128 \text{ М} / n \text{ біт} = 16 \text{ М} / n \text{ байт}$.

$16\text{ М} / n < 16\text{ Кбайт}$
 $n > 1\text{ К}$

Відповідь : список: 16 КБАЙТ;
масив: 16/N МБАЙТ, N> 1 КБАЙТ

Завдання

Текст завдання: Комп'ютер має чотири сторінкових блоки. Час завантаження, час останнього доступу і біти R і M для кожної сторінки наведені нижче (час вказаний у тиках системних годин):

Сторінка	Завантажена	Останнє звертання	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

- а) Яку сторінку вивантажить алгоритм NRU?
- б) Яку сторінку вивантажить алгоритм FIFO?
- в) Яку сторінку вивантажить алгоритм LRU?
- г) Яку сторінку вивантажить алгоритм "друга спроба"?

Розв'язок

Алгоритм NRU вивантажує сторінку, що має найнижчий клас, який вираховується за станом бітів звертання та модифікації – тобто сторінку 2, у якої ці два біти очищені.

Алгоритм FIFO вивантажує сторінку, що має найраніший час завантаження, не звертаючи уваги на те, як давно вона використовувалася. Тобто FIFO вивантажить сторінку 3.

Алгоритм LRU вивантажує сторінку, що має найраніший час звертання, не зважаючи уваги на те, як давно вона завантажена. Тобто LRU вивантажить сторінку 1.

Алгоритм “друга спроба” є версією FIFO, що дозволяє уникнути проблеми витіснення сторінок, що використовуються часто. У найстарішій сторінки вивчається біт R. Якщо він дорівнює 0, сторінка знаходиться у пам'яті довго і не використовується, тобто, вона може бути замінена новою. Якщо біт R дорівнює 1, то він очищується, і сторінка переноситься у кінець списку, а час її завантаження відновлюється на поточний. У нашому випадку найстарішою є сторінка 3, але її біт R дорівнює 1. Для неї буде змінений час завантаження на поточний, та очищений біт звертання. Наступною є сторінка 2, яка має чистий біт звертання, тому вона буде вивантажена.

Відповідь : а) 2; б) 3; в) 1; г) 2

Завдання

Текст завдання: Комп'ютер забезпечує кожен процес 65 536 байт адресного простору, розділеного на сторінки по 4096 байт. Деяка програма має розмір тексту 32 768 байт, розмір даних 16 830 байт і розмір стека 15 870 байт. Чи поміститься ця програма в адресному просторі? А у випадку, якщо розмір сторінки буде 512 байт? Для пояснення відповіді наведіть кількість сторінок, потрібних для розміщення кожного з розділів програми.

Розв'язок

При розмірі сторінки 4096 байт тест програми займе $32\,768/4096 = 8$ сторінок, дані $16\,830/4096 = 5$ сторінок, стек $15\,870/4096 = 4$ сторінки, що разом складатиме $8 + 5 + 4 = 17$ сторінок та перевищує можливості системи ($65\,536 / 4096 = 16$).

При розмірі сторінки 512 байт тест програми займе $32\,768/512 = 64$ сторінки, дані $16\,830/512 = 33$ сторінки, стек $15\,870/512 = 31$ сторінки, що разом складатиме $64 + 33 + 31 = 128$ сторінок та дорівнює можливостям системи ($65\,536 / 512 = 128$).

Відповідь: для розміру 4096 не поміститься ($8+5+4 > 16$), для розміру 512 поміститься ($64 + 33 + 31 = 128$)

Відповідь : а) $8+5+4 > 16$ - НІ; б)
 $64+33+31 = 128$ – ТАК

Завдання

Текст завдання: Розглянемо систему звичайного підкачування, у пам'яті якої містяться вільні ділянки наступних розмірів і в наступному порядку: 10 Кбайт, 4 Кбайт, 20 Кбайт, 18 Кбайт, 7 Кбайт, 9 Кбайт, 12 Кбайт і 15 Кбайт. Який з них буде обраний для успішного задоволення запиту сегмента розміром 12 Кбайт по алгоритму а) "перший придатний"; б) "самий придатний"; в) "самий невідповідний"?

Розв'язок

Алгоритм "перший придатний" починає сканування з першого блоку, та обирає перший блок, розмір якого є рівним або більшим за потрібний. Тобто, якщо у систему надійшов запит на блок розміром 12 Кбайт, то згідно цьому алгоритму буде обраний третій блок розміром 20 Кбайт.

Алгоритм "самий придатний" сканує весь список, та обирає блок, розмір якого є рівним або перевищує потрібний на мінімальну величину. Тобто, якщо у систему надійшов запит на блок розміром 12 Кбайт, то згідно цьому алгоритму буде обраний блок розміром 12 Кбайт.

Алгоритм "самий невідповідний" сканує весь список, та обирає блок, розмір якого перевищує потрібний на максимальну величину. Тобто, якщо у систему надійшов запит на блок розміром 12 Кбайт, то згідно цьому алгоритму буде обраний блок розміром 20 Кбайт (максимальний).

Відповідь : а) 20 Кбайт; б) 12 Кбайт; в) 20 Кбайт

Завдання

Текст завдання: Комп'ютер, чий процеси мають 1024 сторінки у своєму адресному просторі, зберігає таблиці сторінок у пам'яті. На читання слова з таблиці сторінок потрібно 5 нс. Щоб зменшити витрати, у комп'ютері існує буфер швидкого перетворення адреси (TLB), що містить 32 пари (віртуальна сторінка, фізичний сторінковий блок), який може виконати пошук за 1 нс. При якій частоті звертань до пам'яті, успішно реалізованих у TLB, середні витрати будуть нижче 2 нс?

Розв'язок

Позначимо:

N_{TLB} – кількість запитів, що успішно реалізуються у TLB;

t_{TLB} – час реалізації запиту у TLB;

N_{mem} – кількість запитів, що реалізуються у пам'яті;

t_{mem} – час реалізації запиту у пам'яті;

$$t_{cp} = (t_{TLB} * N_{TLB} + (t_{mem} + t_{TLB}) N_{mem}) / (N_{TLB} + N_{mem})$$

Загальну кількість запитів прийемо за 1, тоді $N_{TLB} = 1 - N_{mem}$

Таким чином, $t_{cp} = t_{TLB} + t_{mem} * N_{mem} < 2$ нс

$$N_{mem} < (t_{cp} - t_{TLB}) / t_{mem} = (2 - 1) / 5 = 1/5.$$

Відповідь : більше 80%

Завдання

Текст завдання: Машина має 32-розрядний адресний простір і сторінки розміром 8 Кбайт. Таблиця сторінок цілком підтримується апаратно, на запис у ній приділяється одне 32-розрядне слово. При запуску процесу таблиця сторінок копіюється з пам'яті в апаратуру, копіювання одного слова займає 100 нс. Якщо кожен процес працює протягом 100 мс (включаючи час завантаження таблиці сторінок), яка частка часу процесора втрачається на завантаження таблиці сторінок?

Розв'язок

Одна запис займає 4 байти, всього записів $= 4Г/8К = 2^{19}$, тобто розмір таблиці складає $2^{19} * 4$ байти $= 2^{20}$ слів.

Копіювання одного слова займає 100 нс, отже, копіювання всієї таблиці займає $100 * 2^{20}$ нс $= 52$ мс, що складає 52% часу процесу

Відповідь : 52%.

Завдання

Текст завдання: Один зі способів використовувати безперервні файли на диску і не страждати від "дір", складається в ущільненні диска при кожному вилученому файлі. Оскільки усі файли є безперервними, для копіювання файлу потрібно визначений час на пошук циліндра й обертання диска при зчитуванні файлу, після якого відбувається перенесення даних на повній швидкості. При записі файлу на диск вимагаються аналогічні операції. При часі пошуку циліндра, рівному 5 мс, затримці обертання в 4 мс, швидкості передачі даних 8 Мбайт/с і середньому розмірі файлу 8 Кбайт, скільки буде потрібно часу для ущільнення половини 16-гигабайтного диска?

Розв'язок

Час зчитування дорівнює сумі часів пошуку, затримки обертання та переносу даних:

$$t_{\text{зчит}} = t_{\text{пошук}} + t_{\text{затримки}} + t_{\text{переносу}}.$$

Час переносу $t_{\text{переносу}} = s/v$, де s – середній розмір файлу, v – швидкість переносу даних.

Таким чином час зчитування одного файлу дорівнює:

$$t_{\text{зчит}} = 5 \text{ мс} + 4 \text{ мс} + 8 \text{ Кбайт} / 8 \text{ Мбайт/с} = 10 \text{ мс}$$

Кількість файлів (середнього розміру) на диску розміром 8 Гбайт дорівнює:

$$8 \text{ Гбайт} / 8 \text{ Кбайт} = 10^6 \text{ файлів.}$$

Час ущільнення:

$$t = 10^6 \text{ файлів} * (t_{\text{зчит}} + t_{\text{запису}}) = 10^6 * 20 \text{ мс} = 20 * 10^3 \text{ с.}$$

Відповідь: на ущільнення файлів половини диска розміром 16 Гбайт потрібно 20000 с, або 333 хв, 20 с, або 5 годин 33 хвилини 20 секунд.

Відповідь : $2 * 10^4 \text{ С}$

Завдання

Текст завдання: Яка максимальна швидкість передачі даних для диска, що обертається зі швидкістю 7200 об/хв, при часі переміщення головки на сусідню доріжку, рівному 1 мс? У диска по 200 секторів по 512 байт на кожній доріжці

Розв'язок

Число секторів, що проходить під головкою за 1 мс:

$$N = 7200/60 * 10^{-3} * 200 = 24 \text{ сектора, що складає } 24 * 512 \text{ байт} = 12 \text{ К,}$$

отже швидкість передачі даних складає $12 \text{ К} / 10^{-3} \text{ с} = 12 \text{ Мбайт/с}$

Відповідь : 12 Мбайт/с

Завдання

Текст завдання: Дано гнучкий диск із кроком чергування секторів, рівним 2. На кожній

доріжці диска 8 секторів по 512 байт. Швидкість обертання диска 300 об/хв. Скільки часу буде потрібно, щоб прочитати всі сектори доріжки в правильному порядку, якщо припустити, що головка диска вже знаходиться на потрібній доріжці, а щоб сектор 0 перемістився під головку, потрібно 1/2 оберту диска? а) Чому дорівнює швидкість зчитування даних? б) Проведіть обчислення також для диска без чергування секторів.

Розв'язок

Кількість обертів (N), потрібних для зчитування всіх секторів дорівнює 2,5 (0,5 + 1 (1,2,3,4) + 1 (5,6,7,8)).

$$t_{\text{чт}} = N/v_{\text{чт}} = 2,5 / 300 \text{ об/хв} = 0,5 \text{ с.}$$

Швидкість зчитування даних (диск з чергуванням):

$$v = 8 \text{ секторів} * 512 \text{ байт} / 0,5 \text{ с} = 8192 \text{ байт/с} = 8 \text{ Кбайт/с}$$

Диску без чергування секторів

Для зчитування потрібно 1,5 обороту, отже $t_{\text{чт}} = N/v_{\text{чт}} = 1,5 / 300 \text{ об/хв} = 0,3 \text{ с}$

$$\text{Швидкість: } 8 \text{ секторів} * 512 \text{ байт} / 0,3 \text{ с} = 13,3 \text{ Кбайт/с}$$

Відповідь : а) 8 кбайт/с; б) 13,3 кбайт/с

Завдання

Текст завдання: Чому дорівнює максимальний розмір файлу (в Кбайт), якщо індексний вузол містить 10 полів прямої адресації і по одному полю одинарної, подвійної і потрійної непрямої адресації при розмірі дискових блоків 1024 байт (файлова система є 32-розрядною)?

Розв'язок

$$10 \text{ полів прямої адресації} = 10 * 1024 = 10 \text{ Кбайт};$$

$$1 \text{ поле одинарної непрямої адресації} = 1 \text{ блок} = 1024 \text{ байта} / 4 \text{ байта} = 256 \text{ адрес по } 1024 = 256 \text{ Кбайт};$$

$$1 \text{ поле подвійної непрямої адресації} = 256 * 256 * 1024 = 65\,536 \text{ Кбайт};$$

$$1 \text{ поле потрійної непрямої адресації} = 256 * 256 * 256 * 1024 = 16\,777\,216 \text{ Кбайт};$$

Разом

$$16\,777\,216 \text{ Кбайт} + 65\,536 \text{ Кбайт} + 256 \text{ Кбайт} + 10 \text{ Кбайт} = 16\,843\,018 \text{ Кбайт}$$

Відповідь : 16 843 018 Кбайт

Завдання

Текст завдання: У гнучкого диска 40 циліндрів. Операція пошуку займає 6 мс на циліндр. Якщо не намагатися розмістити блоки файлу близько друг до друга, два логічно послідовних блоки (тобто наступних один за іншим у файлі) виявляться в середньому на відстані 13 циліндрів друг від друга. Однак якщо операційна система намагається поєднувати логічно сусідні блоки в кластери, те середня міжблокова відстань може бути зменшене (наприклад) до двох циліндрів. Скільки знадобиться часу в обох випадках для

зчитування 100-блокового файлу, якщо затримка обертання складає 100 мс, а час переносу одного блоку дорівнює 25 мс?

Розв'язок

Якщо відстань між двома блоками дорівнює 13 циліндрів, то час зчитування 100 блоків дорівнюватиме:

$$100 \text{ блоків} * (6 \text{ мс} * 13 + 100 \text{ мс} + 25 \text{ мс}) = 20,3 \text{ с.}$$

Для відстані у 2 блока:

$$100 \text{ блоків} * (6 \text{ мс} * 2 + 100 \text{ мс} + 25 \text{ мс}) = 13,7 \text{ с.}$$

Відповідь : 20,3 с; 13,7 с

Завдання

Текст завдання: У програмі виконується три потоки, синхронізація яких відбувається за допомогою семафорів:

```
semaphore L = 3, R = 0;
void thread1() {
    for (;;) {
        down(L);
        printf("C");
        up(R);
    }
}

void thread2() {
    for (;;) {
        down(R);
        printf("A");
        printf("B");
        up(L);
    }
}

void thread3() {
    for (;;) {
        down(R);
        printf("D");
    }
}
```

Чи може бути надрукована програмою наступна послідовність символів:
"CABABDDCABCABD"?

Розв'язок

В перший момент часу може виконуватися лише потік 1, тому що другий та третій є заблокованими семафором R. Тобто перший потік друкує "C" та встановлює R=1, L=2. В наступний момент може біти виконаний потік 2, що надрукує "AB", та встановить значення семафорів R=0, L=3. Наступним може бути виконаний тільки потік 1, що друкує літеру "C". Тобто така послідовність символів не може бути надрукована

Відповідь : Ні

Завдання

Текст завдання: В обчислювальній системі зі сторінковою організацією пам'яті час доступу процесора до оперативної пам'яті складає 100 нс, а час доступу до асоціативної пам'яті – 15 нс. Частота попадань до асоціативної пам'яті при звертанні до даних складає 80%. Обчисліть середній час доступу до одиниці даних.

Розв'язок

Позначимо час доступу до оперативної пам'яті як T_o , а час доступу до асоціативної, як T_a , тоді

$T_{\text{середній}} = 0,2(T_a + T_o) + 0,8T_a = 35 \text{ нс}$

Відповідь : 35 нс

Завдання

Текст завдання: Два потоки паралельно виконують код:

Потік 1	Потік 2
1 $x = 1;$	6 $x += 2;$
2 $\text{lock}(\text{mutex});$	7 $\text{lock}(\text{mutex});$
3 $y = y + x;$	8 $y--;$
4 $x = 2;$	9 $x = x - y;$
5 $\text{unlock}(\text{mutex});$	10 $\text{unlock}(\text{mutex});$

На момент запуску потоків, $x = y = 0$ та м'ютекс знаходиться у розблокованому стані. Всі інструкції є атомарними. Перечисліть допустимі пари значень, які можуть бути присвоєні змінним x та y , та відповідні послідовності операцій.

Розв'язок

Визначимо можливі послідовності виконання потоків та значення змінних у цих послідовностях

Номер інструкції	x	y	Номер інструкції	x	y	Номер інструкції	x	y
1	1	0	1	1	0	6	2	0
2-5	2	1	6	3	0	1	1	0
6	4	1	7-10	4	-1	2-5	2	1
7-10	4	0	2-5	2	3	7-10	2	0

Номер інструкції	x	y	Номер інструкції	x	y	Номер інструкції	x	y
1	1	0	6	2	0	6	2	0
6	3	0	7-10	3	-1	1	1	0
2-5	2	3	1	1	-1	7-10	2	-1
7-10	0	2	2-5	2	0	2-5	2	1

Відповідь : $\{4,0\}$: 1,2-5,6,7-10

$\{2,0\}$: 6,1, 2-5, 7-10; 6-10,1-5;

$\{0,2\}$: 1,6,2-,5,7-10;

$\{2,1\}$: 6,1,7-10,2-5;

$\{2,3\}$: 1,6,7-10,2-5.

Завдання

Текст завдання: В обчислювальній системі з сегментною організацією пам'яті з 32-х біт

адресу старші 12 його біт відводяться для номера сегменту.

Яку максимальну кількість сегментів може мати процес? Яким є максимальний розмір сегменту?

Для деякого процесу таблиця сегментів у цій системі має вигляд:

Номер сегмента	Адрес початку сегмента	Довжина сегмента
1	0x00000	0x80000
2	0x20000	0x02000
3	0x10000	0x20000
5	0x30000	0x0f000

Яким фізичним адресам відповідають адреси 0x315678, 0x170201, 0x1300de?

Відповідь :

Максимальна кількість сегментів: 2^{12}

Максимальний розмір сегмента: 2^{20}

Фізичні адреси: 0x25678, 0x70201, 0x300de.

Завдання

Текст завдання: В системі використовуються таблиці сторінок з асоціативною пам'яттю та підкачкою за вимогою. Розрахуйте ефективний час доступу до пам'яті у випадку, якщо час доступу до основної пам'яті складає 100 нс, відсоток влучення в асоціативну пам'ять – 80%, ймовірність сторінкового переривання – 0,001, час обробки сторінкового переривання – 10 мс. Часом доступу до асоціативної пам'яті можна знехтувати.

Розв'язок

Ефективний час може бути розрахований наступним чином: визначимо час, потрібний на виконання 1000 операцій як:

$T = T_{\text{переривання}} + 999 \cdot T_0$,

де $T_{\text{переривання}}$ – час обробки сторінкового переривання;

T_0 – час доступу до основної пам'яті.

$T = 10 \text{ мс} + 999 \cdot 0,2 \cdot 100 \text{ нс} = 10 \text{ мс}$.

Тоді ефективний час можна отримати, розділивши на кількість операцій (1000) = 10 мкс

Відповідь : 10 мкс

Завдання

Текст завдання: Що буде надруковано в результаті виконання наведеного прикладу програми?

```
1  main()
2  {
3      int var = 5;
4      if(fork()) wait(&var);
```

```
5     var++;
6     printf("%d\n",var);
7     return var;
8 }
```

Розв'язок

Оскільки у 4 операторі виникає процес потомок, який наслідує значення var, то він надрукує після виконання внкрементації значення 6. Батьківський процес має дочекатися завершення процесу-нащадку, та отримати статус його завершення, значення якого є довільним цілим числом, що визначається операційною системою. Батьківський процес також інкрементує це число та надрукує його.

Відповідь :

6 та довільне ціле число, що дорівнює статусу завершення Child-процесу + 1.

Завдання

Текст завдання: Скільки копій змінної var буде створено при виконанні наведеного фрагмента програми? Які значення вони отримають в результаті?

```
1  main()
2  {
3      int pid = fork();
4      int var = 5;
5      if(pid == 0) var += 5;
6      else
7      {
8          pid = fork()
9          var += 10;
10         if(pid) var += 5;
11     }
12 }
```

Розв'язок

Оскільки виконується дві операції fork, то ми маємо 3 копії змінної var. Перша і друга копії отримають початкове значення 5, потім для другого процесу виконується строка 5, тобто var = 10, а перший процес виконує ще одну операцію fork. Третя копія наслідує значення var = 5, перший и третій процес виконують var += 10, тобто значення var стає 15, і перший процес ще виконує строчку 10, тобто var стає 20.

Відповідь :

Три копії: перша отримає значення 20, друга: 10, третя: 15.

Завдання

Текст завдання: Одна з редакцій системи UNIX для розрахунку пріоритетів при переключенні процесів застосовує формули:

$\text{decay}(\text{ВЦП}) = \min(\text{пороговий пріоритет}, \text{ВЦП}-10)$;

$\text{пріоритет} = \text{ВЦП}/10 + (\text{базовий рівень пріоритету})$;

де ВЦП – рівень використання центрального процесору.

Заповніть схему диспетчеризації процесів в системі протягом перших п'яти квантів з розрахунком пріоритетів процесів, та вкажіть процес, який буде активним на кожному кванті. Довжина кванта = 1 сек. За цей час таймер надсилає до системи 60 переривань. Кількість процесів у системі дорівнює трьом (А, В, С). В перший квант часу запускається процес А. Всі процеси у системі створюються з пріоритетом 60 (порогове значення).

Розв'язок

На першому кванті запускається процес А. Він є активним, і тому його ВЦП зростає на 60 протягом кванта (величина дорівнює кількості переривань на протязі кванта). Процеси В та С є неактивними, і тому їх ВЦП не змінюється.

По закінченні кванта система розраховує значення пріоритетів.

Для тих процесів, що були неактивними на протязі останнього кванта, розраховується нове значення ВЦП за формулою напіврозпаду:

$\text{decay}(\text{ВЦП}) \text{ В} = \min(\text{пороговий пріоритет}, \text{ВЦП}-10) = \min(60, 0) = 0$;

$\text{decay}(\text{ВЦП}) \text{ С} = \min(\text{пороговий пріоритет}, \text{ВЦП}-10) = \min(60, 0) = 0$;

Та для всіх процесів розраховується нове значення пріоритет з урахуванням ВЦП:

$\text{пріоритет А} = \text{ВЦП}/10 + (\text{базовий рівень пріоритету}) = 60/10 + 60 = 66$;

$\text{пріоритет В} = \text{ВЦП}/10 + (\text{базовий рівень пріоритету}) = 0/10 + 60 = 60$;

$\text{пріоритет С} = \text{ВЦП}/10 + (\text{базовий рівень пріоритету}) = 0/10 + 60 = 60$;

На другому кванті керування отримує процес В, тому що він має вищий пріоритет і є першим в черзі процесів з найвищим пріоритетом. Його ВЦП зростає на 60 протягом кванта. Процеси А та С не є активними.

Розрахунок ВЦП та пріоритетів по закінченні другого кванту:

$\text{decay}(\text{ВЦП}) \text{ А} = \min(\text{пороговий пріоритет}, \text{ВЦП}-10) = \min(60, 60-10) = 50$;

$\text{decay}(\text{ВЦП}) \text{ С} = \min(\text{пороговий пріоритет}, \text{ВЦП}-10) = \min(60, 0) = 0$;

$\text{пріоритет А} = \text{ВЦП}/10 + (\text{базовий рівень пріоритету}) = 50/10 + 60 = 65$;

$\text{пріоритет В} = \text{ВЦП}/10 + (\text{базовий рівень пріоритету}) = 60/10 + 60 = 66$;

$\text{пріоритет С} = \text{ВЦП}/10 + (\text{базовий рівень пріоритету}) = 0/10 + 60 = 60$;

Очевидно, що на третьому кванті керування буде надано процесу С.

Аналогічно розраховуються ВЦП та пріоритети по закінченню третього, четвертого та п'ятого квантів.

Відповідь :

активні процеси виділені в таблиці жирним шрифтом.

Квант	процес А		процес В		процес С	
	пріоритет	ВЦП	пріоритет	ВЦП	пріоритет	ВЦП
1	60	0 -> 60	60	0	60	0
2	66	60->50	60	0->60	60	0
3	65	50->40	66	60->50	60	0->60
4	64	40->100	65	50->40	66	60->50
5	70	100->90	64	40->100	65	50->40