

Контрольная работа 1-01

Вариант 2 (решение)

0. За разговоры с соседом -3 балла за каждый.

1. (6 баллов) В операционных системах, поддерживающих нити исполнения (threads) внутри одного процесса на уровне ядра системы, наряду с блоками управления процессами (PCB) существуют структуры данных для управления нитями - TCB (Thread Control Block). Укажите, какие данные из перечисленных ниже, хранятся, по Вашему мнению, в TCB:

- a. содержимое регистров процессора;
- b. указатель на таблицу страниц памяти;
- c. приоритет нити исполнения;
- d. адрес следующей команды для выполнения (program counter);
- e. указатель стека;
- f. указатель на таблицу открытых файлов.

Решение

- a. Входит. У каждой нити исполнения свой набор значений регистров процессора.
- b. Не входит. Память - это ресурс, выделяемый на процесс, а не на каждый thread в отдельности.
- c. Входит.
- d. Входит. У каждой нити исполнения свой program counter.
- e. Входит. Стеки у разных thread'ов разные.
- f. Не входит. Все системные ресурсы относятся к процессу, а не к отдельным thread'ам.

Оценка:

По одному баллу за правильный ответ на каждый пункт.

2. (3 балла) Кратко сформулируйте разницу между понятиями мультипрограммная вычислительная система и вычислительная система с разделением времени.

Решение

В мультипрограммных вычислительных системах в памяти находится одновременно несколько процессов, получающих возможность исполняться псевдопараллельно, например, когда один процесс ожидает завершения операции ввода-вывода, второй выполняется на процессоре. Системы разделения времени представляют собой частный случай мультипрограммных систем, в которых процессор переключается с процесса на процесс не только на время выполнения операции ввода-вывода, но и просто по прошествии некоторого интервала времени.

3. (12 баллов) Пусть в вычислительную систему поступают пять процессов различной длительности по следующей схеме:

| Номер процесса | Время поступления в систему | Время исполнения |
|----------------|-----------------------------|------------------|
| 1 | 2 | 7 |
| 2 | 6 | 6 |
| 3 | 4 | 4 |
| 4 | 1 | 2 |
| 5 | 0 | 5 |

Вычислите среднее время между стартом процесса и его завершением (*turnaround time*) и среднее время ожидания процесса (*waiting time*) для каждого из трех алгоритмов планирования FCFS (First Come First

Served), RR (Round Robin) и SJF (Short Job First). При вычислениях считать, что процессы не совершают операций ввода-вывода, величину кванта времени принять равной 1, временем переключения контекста пренебречь. Процесс, поступающий в систему, считать готовым к исполнению в момент поступления. Для алгоритма RR принять, что вновь прибывший процесс помещается в начало очереди процессов, готовых к исполнению, и, следовательно, сразу выбирается на исполнение.

Решение

- а. Рассмотрим выполнение процессов в системе для алгоритма FCFS. По вертикали в таблице отложены номера процессов, по горизонтали - моменты времени. Буква И означает состояние исполнения, буква Г - состояние готовности.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | | | Г | Г | Г | Г | Г | И | И | И | И | И | И | И | | | | | | | | | | |
| 2 | | | | | | | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И |
| 3 | | | | | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | | | | | | |
| 4 | | Г | Г | Г | Г | И | И | | | | | | | | | | | | | | | | | |
| 5 | И | И | И | И | И | | | | | | | | | | | | | | | | | | | |

Среднее время между стартом процесса и его завершением: $tt = (12 + 18 + 14 + 6 + 5)/5 = 11$.

Среднее время ожидания: $wt = (5 + 12 + 10 + 4 + 0)/5 = 6.2$.

- б. Рассмотрим выполнение процессов в системе для алгоритма RR:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | | | И | Г | Г | Г | Г | И | Г | Г | Г | И | Г | Г | Г | И | Г | Г | Г | И | Г | И | Г | И |
| 2 | | | | | | | И | Г | Г | Г | И | Г | Г | Г | И | Г | Г | Г | И | Г | И | Г | И | |
| 3 | | | | | И | Г | Г | Г | Г | И | Г | Г | Г | И | Г | Г | Г | И | | | | | | |
| 4 | | И | Г | Г | Г | И | | | | | | | | | | | | | | | | | | |
| 5 | И | Г | Г | И | Г | Г | Г | Г | И | Г | Г | Г | И | Г | Г | Г | И | | | | | | | |

Среднее время между стартом процесса и его завершением: $tt = (22 + 17 + 14 + 5 + 17)/5 = 13.0$.

Среднее время ожидания: $wt = (15 + 11 + 10 + 3 + 12)/5 = 10.2$

- с. Рассмотрим выполнение процессов в системе для алгоритма SJF:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | | | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И | И |
| 2 | | | | | | | Г | Г | Г | Г | Г | И | И | И | И | И | И | | | | | | | |
| 3 | | | | | Г | Г | Г | И | И | И | И | | | | | | | | | | | | | |
| 4 | | Г | Г | Г | Г | И | И | | | | | | | | | | | | | | | | | |
| 5 | И | И | И | И | И | | | | | | | | | | | | | | | | | | | |

Среднее время между стартом процесса и его завершением: $tt = (22 + 11 + 7 + 6 + 5)/5 = 10.2$.

Среднее время ожидания: $wt = (15 + 5 + 3 + 4 + 0)/5 = 5.4$.

Возможно решение для вытесняющего алгоритма с временами $(22+11+7+2+7)/5=9.8$ и $(15+5+3+0+2)/5=5.0$ соответственно.

Оценка:

По 2 балла за каждое правильно вычисленное время.

4. (3 балла) Дайте краткое определение термина "mutual exclusion (взаимоисключение)".

Решение

Под взаимным исключением (mutual exclusion) понимают следующую ситуацию: если один из процессов выполняется в своем критическом участке, то ни один из взаимодействующих с ним процессов не может находиться в своем соответствующем критическом участке.

5. (18 баллов) В вычислительной системе моделируется движение самосвалов от карьера к заводу и обратно по дороге со стареньким мостом. Движение по мосту может осуществляться в обоих направлениях, но на нем не может быть одновременно более трех машин, иначе он рухнет. Каждый самосвал представлен в системе процессом следующей структуры:

Процесс i-й самосвал:

While (1) {

<доехать до моста>
<проехать по мосту>
<доехать до места назначения>

}

Опишите схему безопасного пересечения моста, используя семафоры. Докажите, что Ваша схема удовлетворяет условиям bounded waiting (ограниченного ожидания) и progress. Если несколько процессов ожидают выполнения операции *Signal* над семафором, то считайте, что процесс, который продолжит исполнение после выполнения этой операции, выбирается по принципу FIFO.

Решение

Semaphore breedge_free = 3;

Процесс i-й самосвал:

While (1){

<доехать до моста>
Wait(bridge_free);
<проехать по мосту>
Signal(bridge_free);
доехать до места назначения>

}

Доказательства довольно просты.

Оценка:

За решение с одним семафором - 8 баллов. Если семафоров больше - -2 балла за каждый семафор. За каждое доказательство - 5 баллов.

6. (5 баллов) В вычислительной системе с сегментной организацией памяти из 32-х бит адреса старшие 9 его бит отводятся для номера сегмента.

- а. Какое максимальное количество сегментов может иметь процесс? Каков максимальный размер сегмента?
б. Для некоторого процесса таблица сегментов имеет вид:

| Номер сегмента | Адрес начала сегмента | Длина сегмента |
|----------------|-----------------------|----------------|
| 1 | 0x000000 | 0x100000 |
| 2 | 0x200000 | 0x080000 |
| 3 | 0x100000 | 0x080000 |
| 5 | 0x300000 | 0x300000 |

Каким физическим адресам соответствуют адреса 0x245678, 0x1000006, 0x018b00de?

Решение

- а. На номер сегмента отводится 9 бит, значит, максимальное количество сегментов у процесса $2^{**9} = 512$ сегментов. На смещение внутри сегмента остается $32 - 9 = 23$ бита, поэтому максимальный размер сегмента 2^{**23} байт.

b. 0x245678 -> error (нет сегмента с номером 0), 0x1000006 -> 0x200000 + 0x6 = 0x200006, 0x018b00de -> error (сегмент 3, смещение 0xb00de - больше длины сегмента).

Оценка:

По 1 баллу за каждое значение.

7. (9 баллов) Для некоторого процесса известна следующая строка запросов страниц памяти

1, 2, 0, 3, 2, 1, 3, 0, 5, 2, 7, 1, 2, 7, 0

Сколько ситуаций отказа страницы (*page fault*) возникнет для данного процесса при каждом из трех алгоритмов замещения страниц - FIFO (First Input First Output), LRU (the Least Recently Used), OPT (optimal), если процессу выделено 3 кадра памяти?

Решение

a. Начнем с FIFO:

| | | | | | | | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Строка запросов | 1 | 2 | 0 | 3 | 2 | 1 | 3 | 0 | 5 | 2 | 7 | 1 | 2 | 7 | 0 |
| Страницы в памяти | 1 | 2 | 0 | 3 | 3 | 1 | 1 | 1 | 5 | 2 | 7 | 1 | 1 | 1 | 0 |
| | | 1 | 2 | 0 | 0 | 3 | 3 | 3 | 1 | 5 | 2 | 7 | 7 | 7 | 1 |
| | | | 1 | 2 | 2 | 0 | 0 | 0 | 3 | 1 | 5 | 2 | 2 | 2 | 7 |
| Page fault | + | + | + | + | | + | | | + | + | + | + | | | + |

Количество *page fault*'ов - 10.

b. Для LRU:

| | | | | | | | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Строка запросов | 1 | 2 | 0 | 3 | 2 | 1 | 3 | 0 | 5 | 2 | 7 | 1 | 2 | 7 | 0 |
| Страницы в памяти | 1 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 5 | 5 | 5 | 1 | 1 | 1 | 0 |
| | | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 7 | 7 | 7 | 7 | 7 |
| | | | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 |
| Page fault | + | + | + | + | | + | | + | + | + | + | + | | | + |

Количество *page fault*'ов - 11.

c. Для OPT:

| | | | | | | | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Строка запросов | 1 | 2 | 0 | 3 | 2 | 1 | 3 | 0 | 5 | 2 | 7 | 1 | 2 | 7 | 0 |
| Страницы в памяти | 1 | 2 | 0 | 3 | 3 | 3 | 3 | 0 | 5 | 5 | 7 | 7 | 7 | 7 | 0 |
| | | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Page fault | + | + | + | + | | | | + | + | | + | | | | + |

Количество *page fault*'ов - 8.

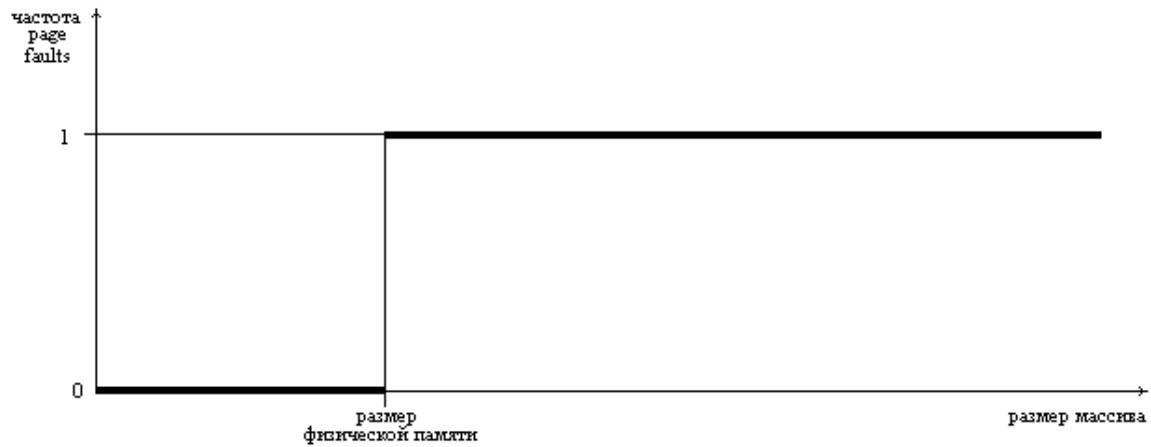
Оценка:

По 3 балла за каждое значение

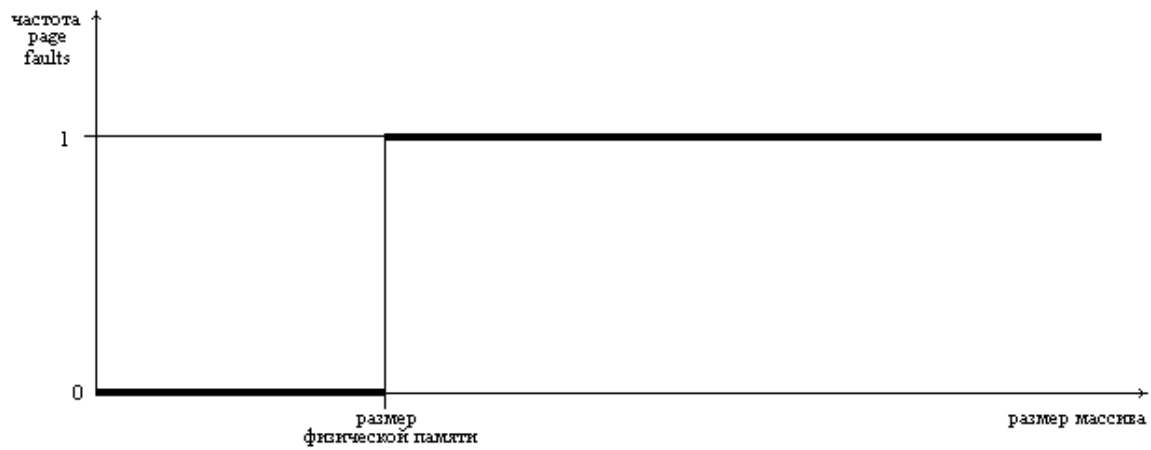
8. (15 баллов) Рассмотрим вычислительную систему со страничной организацией памяти, на которой выполняется одна программа, несколько раз последовательно сканирующая большой одномерный массив данных (это означает, что для массива, занимающего, например, в памяти четыре страницы, строка запроса страниц выглядит следующим образом: 1, 2, 3, 4, 1, 2, 3, 4, ...). Для каждого из трех алгоритмов замещения страниц - FIFO (First Input First Output), LRU (the Least Recently Used) и OPT (OPTimal) нарисуйте график зависимости частоты *page faults* от размеров массива. По оси y откладывается отношение числа *page faults* к длине строки запроса (если массив, занимающий четыре страницы памяти, сканируется 2 раза, то длина строки запроса равна 8), по оси x - размер массива: от размеров, требующих одну страницу, и до размеров, значительно превышающих размер физической памяти. Опишите наиболее характерные точки на графике.

Решение:

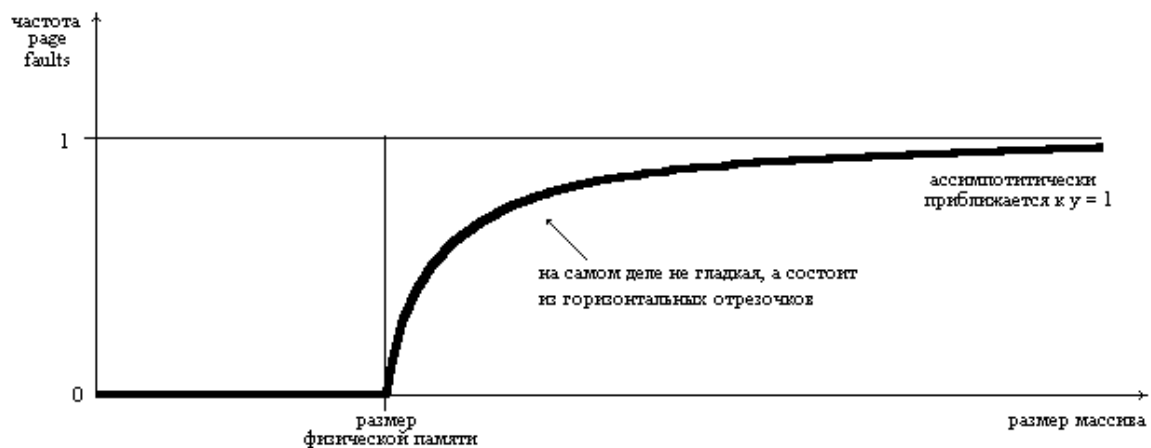
а. Для FIFO:



9. Для LRU:



10. Для OPT:



Оценка:

По 5 баллов за каждый правильный график. Не указано, что точка отхода от 0 значения частоты соответствуют совпадению размеров массива с размером физической памяти - -1балл для каждого графика. Не указано, что кривая для алгоритма OPT асимптотически стремится к прямой $y = 1$ при x , стремящемся к бесконечности, - -1 балл