Контрольная работа 1-00

Вариант 2
(решение)

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

1. (6 баллов) Возможны ли следующие переходы процесса из одного состояния в другое?

a. Из состояния *ожидание* в состояние *готовность*.
b. Из состояния *рождение* в состояние *ожидание*.
c. Из состояния *исполнение* в состояние *готовность*.

Если переход возможен, кратко сформулируйте, когда он происходит. Если невозможен, напишите почему.

Решение:

a. Переход возможен и происходит, когда в системе произошло событие, без которого процесс, находящийся в состоянии *ожидание* не мог продолжить свою работу.

b. Переход невозможен, так как в состояние ожидания может попасть только исполняющийся процесс.

c. Переход возможен и происходит, когда в системе происходит какое-либо прерывание, например, прерывание по таймеру.

Оценка:

По одному баллу за каждый правильный ответ о возможности или невозможности перехода и по 1 баллу за каждое грамотное обоснование.

2. (3 балла) Кратко сформулируйте разницу между программой и процессом.

Решение:

Программа представляет собой статический объект – исполняемый файл. Процесс – это динамический объект, совокупность набора последовательно исполняемых инструкций, связанных с ним ресурсов системы (стеки, основное адресное пространство, выделенные устройства ввода-вывода и т.д.) и их текущего состояния, которая находится под управлением операционной системы.

Таким образом, понятие процесса не включается в понятие программы. С другой стороны понятие программа также не включается в понятие процесса, так как для решения задачи одна программа может породить несколько процессов.

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**	**4**
2	**3**	**10**
3	**6**	**5**
4	**4**	**1**
5	**0**	**4**

Вычислите среднее время между стартом процесса и его завершением (turnaroud 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. Для алгоритма RR принять, что вновь прибывший процесс выбирается для исполнения первым.

Решение:

a. 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	И	И	И	И																
2				Г	Г	Г	Г	Г	И	И	И	И	И	И	И	И	И	И						
3							Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И
4					Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	И					
5	И	И	И	И																				

Среднее время между стартом процесса и его завершением: tt = (6 + 15 + 18 + 15 + 4)/5 = 11.6.
Среднее время ожидания: wt = (2 + 5 + 13 + 14 + 0)/5 = 6.8.

b. 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	И									
2				И	Г	Г	Г	Г	И	Г	Г	Г	И	Г	Г	И	Г	И	Г	И	И	И	И	И
3							И	Г	Г	Г	И	Г	Г	И	Г	Г	И	Г	И					
4					И																			
5	И	И	Г	Г	Г	И	Г	Г	Г	И														

Среднее время между стартом процесса и его завершением: tt = (13 + 21 + 13 + 1 + 10)/5 = 11.8.
Среднее время ожидания: wt = (9 + 11 + 8 + 0 + 6)/5 = 6.8.

c. 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	И	И	И	И															
2				Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И	И	И	И	И	И
3							Г	Г	Г	И	И	И	И	И										
4					И																			
5	И	И	И	И																				

Среднее время между стартом процесса и его завершением: tt = (7 + 21 + 8 + 1 + 4)/5 = 8.2.
Среднее время ожидания: wt = (3 + 11 + 3 + 0 + 0)/5 = 3.4.

Оценка:

По 2 балла за каждое правильно вычисленное время.

4. (3 балла) Дайте краткое определение термина “race condition (состояние гонки)”.

Решение:

Race condition (состояние гонки) — это ситуация, когда результат выполнения процессов, совместно использующих ресурсы, является недетерминированным, т.е. может меняться от запуска к запуску.

5. (18 баллов) Рассмотрим задачу “поможем матушке природе”. Пусть в вычислительной системе существуют процессы, которые “добывают атомы элементов” для последующего их “объединения в молекулы”. Нас будет интересовать “создание молекулы воды”. Для этого в системе существует три процесса: два, добывающих водород, и один, добывающий кислород. Когда в наличии имеется два атома водорода и один кислорода, синтезируется молекула воды. Для решения этой задачи была предложена следующая неправильная схема организации взаимодействия процессов:

```
int nh = 0; /* количество атомов водорода в наличии */
int no = 0; /* количество атомов кислорода в наличии */
```

Процессы, добывающие водород *Ph1* и *Ph2*:

```
while (1) {

        take_h_atom();
        nh = nh +1;

}
```

Процесс, добывающий кислород *Po*:

```
while (1) {

        take_o_atom();
        no = no +1;
        while(nh < 2);
        nh = nh-2;
        no = no-1;
        make_water();

}
```

Укажите, почему она является неправильной, и предложите правильную схему с использованием семафоров. Докажите, что она удовлетворяет условию взаимоисключения

(mutual exclusion).

Решение:

a. Здесь встречается две однотипные ошибки: первая связана с операцией *nh = nh+1* в процессах *Ph1* и *Ph2*, вторая — с операцией *nh = nh-2* в процессе *Po*. Эти операции не являются атомарными. По сути дела они состоят из трех операций: взять значение *nh*, изменить его и положить обратно. Поэтому возможна следующая ситуация:

Пусть nh == 0.
Ph1: взять nh.
Ph2: взять nh.
Ph1: nh = nh+1.
Ph2: nh = nh+1.
Ph1: положить nh.
Ph2: положить nh.

В результате *nh* окажется равным *1*, а не *2*, как должно бы быть. Аналогично рассматривается и ситуация взаимодействия процессов *Phi* и *Po.*

b. Для исправления ситуации воспользуемся семафорами, организовав взаимоисключение на критических участках.

```
int nh = 0; /* количество атомов водорода в наличии */
int no = 0; /* количество атомов кислорода в наличии */

semaphore mutex = 1;
```

Процессы, добывающие водород *Ph1* и *Ph2*:

```
while (1) {

    take_h_atom();
    Wait(mutex);
    nh = nh +1;
    Signal(mutex);

}
```

Процесс, добывающий кислород *Po*:

```
while (1) {

    take_o_atom();
    no = no +1;
    while(nh < 2);
    Wait(mutex);
    nh = nh-2;
    Signal(mutex);
    no = no-1;
    make_water();

}
```

c. Докажем выполнение условия mutual exclusion (взаимоисключения). Доказательство индукцией по числу входов в критические участки. Для первого входа в критический

участок один из процессов успешно преодолел операцию *Wait,* первоначальное значение семафора было *1,* значит, стало *0* и остальные процессы не могут войти в критический участок, пока этот процесс не покинет его. Пусть доказано для *n* входов, докажем для *n+1* – го. На *n*-м входе внутри критического участка находился лишь один процесс. Значит, значение семафора равнялось *0.* Этот процесс произвел одну операцию *Signal* при выходе из критического участка, что позволило только одному из процессов войти в свой критический участок на *n+1* – м входе, а остальные должны ожидать, пока он покинет критический участок и выполнит операцию *Signal*.

Оценка:

По 6 баллов за каждый пункт.

6. **(5 баллов) В вычислительной системе с сегментной организацией памяти из 32-х бит адреса старшие 10 его бит отводятся для номера сегмента.**

   a. **Какое максимальное количество сегментов может иметь процесс? Каков максимальный размер сегмента?**

   b. **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, 0x500001, 0x00c700de?

Решение:

1. На номер сегмента отводится 10 бит, значит, максимальное количество сегментов у процесса 2**10 = 1024 сегмента. На смещение внутри сегмента остается 32 - 10 = 22 бит, поэтому максимальный размер сегмента 2**22 байт.

2. 0x245678 —> error (нет сегмента с номером 0), 0x500001 —> error (сегмент 1, смещение 0x100001 — больше, чем длина сегмента), 0x00c700de —> (сегмент 3, смещение 0x700de — меньше, чем длина сегмента) —>0x100000+0x700de = 0x1700de.

Оценка:

По 1 баллу за каждое значение.

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

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

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

Решение:

a. Начнем с FIFO:

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

Количество page fault'ов — 9.

b. Для LRU:

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

Количество page fault'ов — 11.

c. Для OPT:

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

Количество page fault'ов — 8.

Оценка:

По 3 балла за каждое значение

8. **(15 баллов) Ваш научный руководитель обратился к Вам за помощью для составления конфигурации приобретаемого компьютера. Вы можете потратить $1500 на приобретение следующих компонент:**

Компонента	Время доступа	Минимальный размер	Стоимость
TLB	**10 наносекунд**	**4 входа**	**$20 за вход**
Оперативная память	**200 наносекунд**	**16 Mb**	**$2 за Mb**
Жесткий диск	**10 миллисекунд**	**2 Gb**	**$0.20 за Mb**

Размер страницы памяти в системе фиксирован и составляет 16 Kb. Предполагается, что в системе будут одновременно работать 3-4 приложения, каждое из которых требует не более

64 Mb памяти и имеет рабочий объем памяти (общий размер рабочего набора страниц) 256 Kb. Сформулируйте Ваши предложения руководителю по конфигурации системы.

Решение:

Начнем с диска. Так как диск наиболее медленная компонента системы, мы берем наименьший размер. 2 Gb = 2048 Mb или $410. У нас осталось $1090. Минимальное количество входов для TLB, которое необходимо для отображения рабочего набора страниц составляет 256Kb / 16Kb = 16 входов или $320. У нас остается $770. С этими деньгами мы могли бы приобрести 385Mb памяти, однако нам нужно только 4 * 64Mb = 256 Mb памяти или $512. Остаток $258 потратим еще на 12 TLB входов ($240). Увеличивая количество TLB, мы до некоторой степени улучшаем производительность системы при смене рабочих наборов страниц. Дальнейшее увеличение количества TLB потребовало бы уменьшения оперативной памяти, что привело бы к возникновению paging'а.

Оценка:

За правильный выбор для каждой компоненты по 3 балла, исходя из таблицы:

Компонента/ баллы	3 балла	2 балла	1 балл	0 баллов
TLB	29 >= TLB>=16	32>=TLB>29	TLB>32	TLB<16
Память	>=256	256>Mem>=128	128>Mem>=64	<64
Диск	2Gb	>2Gb при закупке TLB и памяти на 3 балла	>2Gb	<2Gb

За выбор TLB исходя из рабочего объема памяти + 3 балла. За разумное обоснование конфигурации + 3 балла.