

Контрольная работа 1-05

Вариант 12 (решения)

За разговоры с соседом -3 балла за каждый разговор.

1. (14 баллов) Рассмотрим однопроцессорную вычислительную систему с объемом оперативной памяти 200 Мб, в которой используется схема организации памяти с динамическими (переменными) разделами. Для долгосрочного планирования процессов в ней применен алгоритм SJF. В систему поступают пять заданий с различной длительностью и различным объемом занимаемой памяти по следующей схеме:

Номер задания	Момент поступления в очередь заданий	Время исполнения (CPU burst)	Объем занимаемой памяти
1	0	3	80 Мб
2	2	4	50 Мб
3	3	5	60 Мб
4	4	2	80 Мб
5	5	1	10 Мб

Вычислите среднее время между стартом задания и его завершением (turnaround time) и среднее время ожидания (waiting time) для следующих комбинаций алгоритмов краткосрочного планирования и стратегий размещения процессов в памяти:

- RR (Round Robin) и worst fit (наименее подходящий);
- RR и best fit (наиболее подходящий);
- невывесняющий SJF (Short Job First) и worst fit;
- невывесняющий SJF и best fit.

При вычислениях считать, что процессы не совершают операций ввода-вывода, величину кванта времени принять равной 4. Временами переключения контекста, рождения процессов и работы алгоритмов планирования пренебречь. Освобождение памяти, занятой процессами, происходит немедленно по истечении их CPU burst. Краткосрочное планирование осуществляется после рождения новых процессов в текущий момент времени. Для алгоритма RR принять, что родившиеся процессы добавляются в **САМЫЙ** конец очереди готовых процессов (**ПОСЛЕ** процесса, перешедшего в состояние *готовность* из состояния *исполнение* в это время).

Решение:

- a. Рассмотрим выполнение процессов в системе для алгоритма RR и стратегии worst fit. По вертикали в таблице отложены номера процессов, по горизонтали — промежутки времени. Столбец 0 соответствует временному интервалу от 0 до 1. Буква И означает состояние исполнения, буква Г — состояние готовности, буква О — ожидание в очереди заданий. Под таблицей приведено распределение памяти, а еще ниже — содержимое очереди заданий.

[illegible][illegible]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
				P ₄	P ₄	P ₄	P ₄	P ₄	P ₄	P ₄	P ₄			

Среднее время между стартом задания и его завершением: $tt = (3 + 5 + 10 + 11 + 7)/5 = 7.2$.

Среднее время ожидания: $wt = (0 + 1 + 5 + 9 + 6)/5 = 4.2$.

б. Рассмотрим выполнение процессов в системе для алгоритма RR и стратегии best fit.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	И	И	И												
2			Г	И	И	И	И								
3				Г	Г	Г	Г	И	И	И	И	Г	Г	Г	И
4					Г	Г	Г	Г	Г	Г	Г	И	И		
5						Г	Г	Г	Г	Г	Г	Г	Г	И	

80 P ₁	80 P ₁	80 P ₁	80	80 P ₄	80 P ₄	80 P ₄	80 P ₄	80 P ₄	80 P ₄	80 P ₄	80 P ₄	80 P ₄	80 P ₄	130	130
			50 P ₂	50 P ₂	50 P ₂	50 P ₂	50 P ₂	50	50	50	50	50	50		
120	120		70	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃
				10	10	10 P ₅	10 P ₅	10 P ₅	10 P ₅	10 P ₅	10 P ₅	10 P ₅	10 P ₅	10 P ₅	10

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

С₁ Среднее время между стартом задания и его завершением: $tt = (3 + 5 + 12 + 9 + 9)/5 = 7.6$.

Среднее время ожидания: $wt = (0 + 1 + 7 + 7 + 8)/5 = 4.6$.

с. Рассмотрим выполнение процессов в системе для невытесняющего алгоритма SJF и стратегии worst fit.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	И	И	И												
2			Г	И	И	И	И								
3				Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И
4					О	О	О	О	И	И					
5						Г	Г	И							

80 P ₁	80 P ₁	80 P ₁	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃
			20	20	20	20									
							70								
			50 P ₂	50 P ₂	50 P ₂	50 P ₂	50 P ₂	80 P ₄	80 P ₄						
120	120														
			70	70	70										
						10 P ₅	10 P ₅	10 P ₅							
						60	60	60	60	60					
											140	140	140	140	140

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
					P ₄	P ₄	P ₄	P ₄							

Среднее время между стартом задания и его завершением: $tt = (3 + 5 + 12 + 6 + 3)/5 = 5.8$

Среднее время ожидания: $wt = (0 + 1 + 7 + 4 + 2)/5 = 2.8$.

d. Рассмотрим выполнение процессов в системе для невытесняющего алгоритма SJF и стратегии best fit.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	И	И	И												
2			Г	И	И	И	И								
3				Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И
4					Г	Г	Г	Г	И	И					
5						Г	Г	И							

80 P ₁	80 P ₁	80 P ₁	80	80 P ₄	80 P ₄	80 P ₄	80 P ₄	80 P ₄	80 P ₄	130	130	130	130	130
		50 P ₂	50 P ₂	50 P ₂	50 P ₂	50 P ₂	50	50	50					
120	120	70	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃	60 P ₃
			10	10	10 P ₅	10 P ₅	10 P ₅	10	10	10	10	10	10	10

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Среднее время между стартом задания и его завершением: $tt = (3 + 5 + 12 + 6 + 3)/5 = 5.8$.

Среднее время ожидания: $wt = (0 + 1 + 7 + 4 + 2)/5 = 2.8$.

Оценка:

За каждый алгоритм со стратегией — по 3 балла. Если времена нахождения в очереди заданий включены в подсчет времен — еще 2 балла на всю задачу

- (12 баллов) У пустой бочки спит медведь. К бочке прилетают пчелы и кидают в бочку по капле меда. Одновременно положить в бочку мед две пчелы не могут. Бочка вмещает N капель. Пчела, заполнившая бочку до конца, жалит медведя. Медведь просыпается, съедает мед, отгоняя пчел, и снова засыпает. После этого процесс повторяется. Используя семафоры Дейкстры и разделяемые переменные, постройте корректную модель происходящего, описав поведение каждой из пчел и медведя с помощью отдельных процессов.

Решение:

Заводим 2 семафора lock_barrel (для ограничения доступа к бочке) и may_eat (для активизации медведя) и разделяемую переменную Nportion (для количества капель).

Semaphore lock_barrel = 1, may_eat = 0;

Shared int Nportion = 0;

Для пчел

```
While(1){  
    P(lock_barrel);  
    Nportion++;  
    Положить мед;  
    if(Nportion == N) { Ужалить; V(may_eat);}  
    else V(lock_barrel);  
    Улететь за медом  
}
```

Для медведя

```
While(1){  
    P(may_eat);  
    Съесть мед;  
    Nportion = 0;  
    V(lock_barrel);  
    Лечь спать;  
}
```

Оценка:

Грубые ошибки: нет взаимoisключения, тупиковые ситуации, убитые за попытку положить мед не вовремя пчелы, умерший от голода медведь — -8 баллов, средней тяжести: циклы ожидания, прохождение пчелами критических участков без совершения разумных действий — -4 балла. Полный балл только за полностью правильный ответ.

3. (6 баллов) В вычислительной системе с сегментно-страничной организацией памяти и 32-х битовым адресом максимальный размер сегмента составляет 4 Mb, а размер страницы памяти 512 Kb. Для некоторого процесса в этой системе таблица сегментов имеет вид:

Номер сегмента	Длина сегмента
0	0x180000
1	0x080000

Таблицы страниц, находящихся в памяти, для сегментов 0 и 1 приведены ниже:

Сегмент 0	
Номер страницы	Номер кадра (десятичный)
0	18
3	0

Сегмент 1	
Номер страницы	Номер кадра (десятичный)
0	32
1	63

Каким физическим адресам соответствуют логические адреса: 0x000f0236, 0x00470111, 0x00502005?

Решение:

4 Mb — это 2^{22} байт, т.е. под номер сегмента в логическом адресе отводится 10 бит, а 22 бита — под смещение внутри сегмента. Размер страницы 512 Kb — это 2^{19} байт, т.е. из смещения внутри сегмента 19 бит отводится под смещение внутри страницы, а 3 бита — под номер страницы.

0x000f0236 —> сегмент 0, смещение 0xf0236 —> сегмент 0, страница 1, смещение 0x00070236 —> **error**,

0x00470111 —> сегмент 1, смещение 0x070111 —> сегмент 1, страница 0, смещение 0x00070236 —> кадр 32, смещение 0x00070236 —> **0x01070236**,

0x00502005 —> сегмент 1, смещение 0x102005 —> смещение больше размера сегмента —> **error**.

Оценка:

По 2 балла за адрес:

4. (6 баллов) Ответьте на следующие вопросы:

- Какие из следующих схем организации памяти не могут быть использованы для организации виртуальной памяти: страничная организация, сегментная организация, организация динамических разделов? Обоснуйте свой ответ.
- Что такое ассоциативная память? Для чего она применяется в вычислительных системах?

Решение:

- Схема с динамическими разделами — это единственная схема организации из перечисленных в условии, которая не может быть использована для организации виртуальной памяти, так как при этой схеме исполняющийся процесс должен полностью находиться в оперативной памяти, и, следовательно, его логическое адресное пространство не может превышать по своим размерам физическое адресное пространство.
- В соответствии с принципом локальности, большинство процессов в системе со страничной организацией памяти в течение некоторого промежутка времени делают обращения только к небольшому числу страниц. Для ускорения доступа к данным компьютер снабжается аппаратным устройством — быстрой кэш-памятью, хранящей необходимую на данный момент часть таблицы страниц. Это устройство называется ассоциативной памятью, иногда также употребляют термин ассоциативные регистры (TLB). Одна запись в ассоциативной памяти содержит информацию про одну страницу логического адресного пространства: ее номер, адрес соответствующего физического кадра и ее атрибуты.

Оценка:

За каждый пункт предполагается по 3 балла.