**Контрольная работа 1-05**

**Вариант 11 (решения)**

За разговоры с соседом -3 балла за каждый разговор.

1. (14 баллов) Рассмотрим однопроцессорную вычислительную систему с объемом оперативной памяти 200 Mb, 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 Mb |
| 2 | 2 | 4 | 50 Mb |
| 3 | 3 | 5 | 60 Mb |
| 4 | 4 | 2 | 80 Mb |
| 5 | 5 | 1 | 10 Mb |

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:

1. RR (Round Robin) и first fit (первый подходящий);
2. RR и best fit (наиболее подходящий);
3. вытесняющий SJF (Short Job First) и first fit;
4. вытесняющий SJF и best fit.

При вычислениях считать, что процессы не совершают операций ввода-вывода, величину кванта времени принять равной 2. 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:***

* 1. Рассмотрим выполнение процессов в системе для алгоритма RR и стратегии first 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.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 80 P1 | 80 P1 | 80 P1 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 |
| 20 | 20 | 10 P5 | 10 P5 | 10 P5 | 10 P5 | 10 P5 | 10 | 10 | 10 | 10 | 140 |
| 10 | 10 | 10 | 10 | 80 P4 | 80 P4 | 80 P4 | 80 P4 | 80 P4 |
| 120 | 120 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 |
| 70 | 70 | 70 | 70 | 70 | 70 | 70 |
| 50 | 50 | 50 | 50 | 50 |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|  |  |  |  | P4 | P4 | P4 | P4 | P4 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Среднее время между стартом задания и его завершением: tt = (3 + 7 +12 + 10 + 5)/5 = 7.4.  
Среднее время ожидания: wt = (0 + 3 + 7 + 8 + 4)/5 = 4.4.

* 1. 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 | Г | Г | Г | Г | И | И |  |  |  |  |
| 3 |  |  |  | Г | Г | И | И | Г | Г | Г | Г | Г | И | И | И |
| 4 |  |  |  |  | Г | Г | Г | И | И |  |  |  |  |  |  |
| 5 |  |  |  |  |  | Г | Г | Г | Г | Г | Г | И |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 80 P1 | 80 P1 | 80 P1 | 80 | 80 P4 | 80 P4 | 80 P4 | 80 P4 | 80 P4 | 80 | 80 | 130 | 130 | 130 | 130 |
| 120 | 120 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 |
| 70 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 |
| 10 | 10 | 10 P5 | 10 P5 | 10 P5 | 10 P5 | 10 P5 | 10 P5 | 10 P5 | 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 + 9 +12 + 5 + 7)/5 = 7.2.  
Среднее время ожидания: wt = (0 + 5 + 7 + 3 + 6)/5 = 4.2.

* 1. Рассмотрим выполнение процессов в системе для вытесняющего алгоритма SJF и стратегии first fit.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 80 P1 | 80 P1 | 80 P1 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 |
| 20 | 20 | 10 P5 | 20 | 20 | 80 P4 | 80 P4 | 80 P4 | 80 P4 | 140 | 140 | 140 |
| 10 |
| 120 | 120 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P |
| 70 | 70 | 70 | 70 | 70 | 70 |
| 60 | 60 | 60 | 60 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|  |  |  |  | P4 | P4 | P4 | P4 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Среднее время между стартом задания и его завершением: tt = (3 + 6 + 12 + 8 + 1)/5 = 6.0  
Среднее время ожидания: wt = (0 + 2 + 7 + 6 + 0)/5 =3.0.

* 1. Рассмотрим выполнение процессов в системе для вытесняющего 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 | Г | Г | Г | И | И | И |  |  |  |  |  |
| 3 |  |  |  | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И |
| 4 |  |  |  |  | И | И |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  | Г | И |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 80 P1 | 80 P1 | 80 P1 | 80 | 80 P4 | 80 P4 | 80 | 80 | 80 | 80 | 130 | 130 | 130 | 130 | 130 |
| 120 | 120 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 | 50 P2 |
| 70 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 | 60 P3 |
| 10 | 10 | 10 P5 | 10 P5 | 10 | 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 + 8 + 12 + 2 + 2)/5 = 5.4.  
Среднее время ожидания: wt = (0 + 4 + 7 + 0 + 1)/5 = 2.4.

*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

1. (12 баллов) В диком каннибальском племени вокруг котла с пищей спят дикари и повар. Изначально в котле находится N порций мяса. Дикари по очереди просыпаются, берут из котла порцию мяса, съедают его и засыпают снова. Дикарь, не обнаруживший мяса в котле, будит повара. Повар находит добычу и снова готовит N порций, не подпуская никого к котлу во время приготовления, после чего тоже засыпает. Используя классические очереди сообщений и разделяемые переменные, постройте корректную модель происходящего, описав поведение каждого из дикарей и повара с помощью отдельных процессов. Классические очереди сообщений используют примитивы **send(A,message)** и **receive(A,message)**, где A - имя очереди сообщений.

***Решение:***

Заводим 2 очереди сообщений A (для дикарей) и B (для повара) и разделяемую переменную Nportion (для количества порций).

Shared int Nportion = N;

|  |  |
| --- | --- |
| Для дикарей  While(1){  receive(A, message);  if(Nportion == 0)  {Разбудить повара; send(B,message);}  **else{**  **Взять порцию; Nportion--;**  send(A, message); Съесть порцию; Поспать;  }  } | Для повара  send(A, message);  While(1){  receive(B, message);  Найти добычу и приготовить еду;  Nportion = N;  send(A, message);  Лечь спать;  } |

*Оценка:*

Грубые ошибки: нет взаимоисключения, тупиковые ситуации, убитые за попытку взять пищу не вовремя дикари — -8 баллов, средней тяжести: циклы ожидания, прохождение дикарями критических участков без совершения разумных действий — -4 балла. Полный балл только за полностью правильный ответ.

1. (6 баллов) В вычислительной системе с сегментно-страничной организацией памяти и 32-х битовым адресом максимальный размер сегмента составляет 2 Mb, а размер страницы памяти 256 Kb. Для некоторого процесса в этой системе таблица сегментов имеет вид:

|  |  |
| --- | --- |
| Номер сегмента | Длина сегмента |
| 0 | 0x7e000 |
| 1 | 0x080000 |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сегмент 0 | |  | Сегмент 1 | |
| Номер страницы | Номер кадра  (десятичный) |  | Номер страницы | Номер кадра  (десятичный) |
| 0 | 0 |  | 0 | 63 |
| 1 | 12 |  | 2 | 32 |

Каким физическим адресам соответствуют логические адреса: 0x0007f123, 0х002b3312, 0x00502005?

***Решение:***

2 Mb — это 221 байт, т.е. под номер сегмента в логическом адресе отводится 11 бит, а 21 бит — под смещение внутри сегмента. Размер страницы 256 Kb — это 218 байт, т.е. из смещения внутри сегмента 18 бит отводится под смещение внутри страницы, а 3 бита — под номер страницы.

**0x0007f123** —> сегмент 0, смещение 0x07f123 —> смещение больше длины сегмента —>**error**,   
**0x002b3312** —> сегмент 1, смещение 0x0b3312 —> сегмент 1, страница 2, смещение 0x00033312 — > кадр 32, смещение 0x00033312 —> **0x00833312**,   
**0x00502005** —> сегмент 2, смещение 0x102005 —>**error**.

*Оценка:*

По 2 балла за адрес:

1. (6 баллов) Ответьте на следующие вопросы:
2. Что понимается под термином spooling (спулинг)?
3. Что понимается под термином mutual exclusion (взаимоисключение)? Достаточно ли организации взаимоисключений для корректной работы взаимодействующих процессов?

***Решение:***

1. На основании знаний, данных на лекции к моменту написания контрольной, ответ должен выглядеть примерно так: «Одновременное выполнение на одном компьютере вычислений на центральном процессоре для одного процесса и реальных операций ввода-вывода на медленных устройствах, требующих монопольного использования (таких как принтер, устройства ввода с перфокарт и т.д.) для другого процесса, получило название spooling (от SPOOL — Simultaneous Peripheral Operations On Line)».
2. Под взаимоисключением (mutual exclusion) понимают следующую ситуацию: если один из процессов исполняется в своем критическом участке, то ни один из взаимодействующих с ним процессов не может находиться в своем соответствующем критическом участке. Более общо: взаимоисключение — это обеспечение эксклюзивного доступа к какому-либо ресурсу. Одних взаимоисключений недостаточно для организации корректной работы взаимодействующих процессов. Если очередность доступа к ресурсу не важна, то необходимо еще выполнение условий ограниченного ожидания и прогресса. Если очередность доступа имеет значение, то требуется дополнительно взаимная синхронизация работы процессов.

*Оценка:*

За каждый пункт предполагается по 3 балла.