**Задачи по теме «МОДУЛИ ОС»**

1. Известно, что программа А выполняется в монопольном режиме за 10 минут, а программа В — за 20 минут, то есть при последовательном выполнении они требуют 30 минут. Составьте диаграммы выполнения этих задач в монопольном и мультипрограммных режимах и оцените Т —время выполнения обеих этих задач в режиме мультипрограммирования. Ответ обоснуйте  
     
   **Решение:** В монопольном режиме выполнения программа А выполняется за 10 минут, а программа В - за 20 минут. При последовательном выполнении этих программ общее время выполнения составляет 30 минут.   
      
    В мультипрограммном режиме выполнения задачи выполняются параллельно, поэтому время выполнения общей программы будет меньше, чем 30 минут. Чтобы оценить время выполнения в мультипрограммном режиме, рассмотрим две диаграммы выполнения задач в мультипрограммном и монопольном режимах.   
     
    Диаграмма выполнения задач в мультипрограммном режиме:

На этой диаграмме задачи A и B выполняются параллельно. Время выполнения общей программы будет равно времени выполнения самой долгой задачи, то есть 20 минут.   
  
 Диаграмма выполнения задач в монопольном режиме:   
  
  
  
  
  
  
  
 На этой диаграмме задача A выполняется первой, а затем начинается выполнение задачи B. Время выполнения общей программы будет равно времени выполнения самой долгой задачи, увеличенному на время выполнения более короткой задачи, то есть 20 + 10 = 30 минут.   
  
 Таким образом, при оптимальном порядке выполнения задач в мультипрограммном режиме, время выполнения общей программы будет равно 20 минутам.

**2.**Система устраняет свободные участки памяти с помощью уплотнения. Если предположить, что множество свободных участков и множество сегментов, данных распределено случайно, а время для чтения и записи 32 – разрядного слова в памяти равно 10 нс, то, сколько времени займёт уплотнение 128 Мбайт памяти в худшем случае? Приведите расчёт и опишите механизм уплотнения памяти.  
  
**Решение:**  
 Механизм уплотнения памяти (Defragmentation) заключается в объединении свободных участков памяти в единый блок, чтобы создать большой непрерывный блок свободной памяти, который может быть использован для выделения новых сегментов данных.

Для расчета времени уплотнения памяти в худшем случае, необходимо оценить количество свободных участков памяти, которые нужно объединить. Допустим, что в 128 Мбайт памяти имеется 100000 свободных участков размером 128 байт каждый. Это означает, что в худшем случае нужно объединить все эти участки в один большой блок.

Для объединения этих участков необходимо выполнить следующие действия:

Прочитать все данные из каждого свободного участка памяти и сохранить их во временной области памяти.

Записать данные обратно в память в одном большом блоке, начиная с самого начала памяти.

Предположим, что для чтения и записи каждого 32-разрядного слова требуется 10 нс. Тогда для чтения и записи каждого свободного участка памяти размером 128 байт потребуется 4096 таких операций. Для 100000 участков это будет составлять 409600000 операций чтения и записи.

Таким образом, общее время, затраченное на уплотнение памяти в худшем случае, будет равно:

409600000 операций \* 10 нс/операция = 4096000000 нс = 4.1 секунды

Таким образом, уплотнение памяти может занять достаточно много времени, особенно если в памяти имеется множество свободных участков. Поэтому регулярное выполнение уплотнения памяти может помочь улучшить производительность системы, уменьшив фрагментацию памяти.

**3.**Компьютер имеет 32- разрядное адресное пространство и страницы размером 8 Кбайт. Таблица страниц целиком поддерживается аппаратно, на запись в ней отводится одно 32 –разрядное слово. При запуске процесса таблица страниц копируется из памяти в аппаратуру, одно слово требует 10нс. Если каждый процесс работает в течение 100мс (включая время загрузки таблицы страниц), какая доля времени процессора жертвуется на загрузку таблицы страниц? Приведите расчёт и опишите страничное распределение виртуальной памяти.  
  
**Решение:** 32-разрядное адресное пространство означает, что адреса памяти могут быть выражены в 32-разрядном двоичном коде, т.е. существует 2 в 32-й степени (4 294 967 296) возможных адресов. Размер страницы равен 8 Кбайт, что составляет 2 в 13-й степени (8192) байт. Таким образом, количество страниц в адресном пространстве равно 2 в 32-й степени / 2 в 13-й степени = 2 в 19-й степени = 524 288 страниц.  
  
 Таблица страниц хранит информацию о том, какие страницы находятся в физической памяти, а какие загружены на диск. Каждая запись в таблице страниц содержит флаги, которые помогают понять, как использовать эту страницу (например, был ли она изменена с момента ее последней загрузки, можно ли ее использовать в различных режимах доступа и т.п.).  
  
 Время, которое требуется для копирования таблицы страниц в аппаратуру, равно количеству слов в таблице, умноженному на время копирования одного слова. Количество слов в таблице равно количеству записей в таблице страниц, умноженному на количество слов в каждой записи (одно 32-разрядное слово). Количество записей в таблице страниц равно количеству страниц в адресном пространстве (524 288).  
  
 Таким образом, количество слов в таблице страниц равно 524 288 **1 = 524 288, а время копирования одного слова равно 10нс. Тогда общее время, которое требуется для загрузки таблицы страниц в аппаратуру, составляет 524 288** 10нс = 5.24288мс.  
  
 Время работы процесса равно 100мс, а время, затрачиваемое на загрузку таблицы страниц, равно 5.24288мс. Тогда доля времени, затрачиваемая на загрузку таблицы страниц, равна 5.24288мс / 100мс = 0.0524, т.е. около 5%.  
  
 Страничное распределение виртуальной памяти означает, что адресное пространство процесса разделено на страницы фиксированного размера, а каждая страница может находиться либо в физической памяти, либо на диске. Когда процесс обращается к адресу в виртуальном адресном пространстве, аппаратура переводит этот адрес в соответствующий адрес физической памяти. Если страница находится на диске, она загружается в память, а если страница уже находится в памяти, процессор может использовать ее напрямую.  
  
 Страничное распределение виртуальной памяти позволяет эффективно использовать физическую память и много задействовать дисковое пространство, так как можно загружать и выгружать страницы по мере необходимости. Однако это также создает некоторые накладные расходы из-за необходимости копировать таблицу страниц в аппаратуру при каждом запуске процесса и проверять флаги для каждой страницы при каждом обращении к виртуальному адресу.

1. В системе со страничной организацией памяти (размер страницы 214байт) происходит обращение по виртуальному адресу 012356. Преобразуйте этот адрес в физический, если в таблице страниц данного процесса задана следующая информация

|  |  |
| --- | --- |
| **Номер виртуальной страницы** | **Номер физической страницы** |
| **0000** | **0101** |
| **0001** | **0010** |
| **0010** | **0011** |
| **0011** | **0000** |

Поясните страничное распределение памяти.  
  
**Решение:** Для преобразования виртуального адреса в физический необходимо разбить его на номер страницы и смещение внутри страницы. В данном случае размер страницы равен 2^14 байт, поэтому первые 2 бита виртуального адреса отвечают за номер страницы, а оставшиеся 14 бит за смещение внутри страницы.  
   
 Таким образом, для адреса 012356:  
   
 - Номер виртуальной страницы: 01 23 (первые 2 бита)   
 - Смещение внутри страницы: 56 (оставшиеся 14 бит)   
  
 Согласно заданию, виртуальная страница 01 соответствует физической странице 0010, поэтому номер физической страницы также будет равен 01 10 (первые 2 бита), а смещение внутри страницы останется тем же - 56.   
  
 Таким образом, физический адрес будет равен 0110 0056.   
  
 Страничное распределение памяти представляет собой метод организации виртуальной памяти, при котором программа работает с виртуальными адресами, которые затем преобразуются в физические адреса с помощью таблицы страниц. Каждая страница имеет свой номер, который используется для определения соответствующей физической страницы. Это позволяет эффективно использовать доступную память, разбивая ее на страницы и загружая в оперативную память только те страницы, которые необходимы для выполнения программы.

**5**.Вычислить номер виртуальной страницы и смещение для виртуальных адресов 1230005, 3274893, если размер страницы равен 4 Кбайт Приведите расчёт и опишите страничное распределение виртуальной памяти

**Решение:**

Для вычисления номера виртуальной страницы и смещения необходимо знать размер страницы и виртуальный адрес. В данном случае размер страницы равен 4 Кбайт.

Для вычисления номера виртуальной страницы и смещения используются следующие формулы:

Номер виртуальной страницы = Виртуальный адрес / Размер страницы

Смещение = Виртуальный адрес % Размер страницы

Давайте применим эти формулы к каждому из виртуальных адресов:

Для виртуального адреса 1230005:

Номер виртуальной страницы = 1230005 / 4096 ≈ 300

Смещение = 1230005 % 4096 = 1429

Таким образом, для виртуального адреса 1230005 номер виртуальной страницы равен 300, а смещение равно 1429.

Для виртуального адреса 3274893:

Номер виртуальной страницы = 3274893 / 4096 ≈ 800

Смещение = 3274893 % 4096 = 357

Таким образом, для виртуального адреса 3274893 номер виртуальной страницы равен 800, а смещение равно 357.

Страничное распределение виртуальной памяти определяется по номеру виртуальной страницы. Каждая страница фиксированного размера отображается на определенную область физической памяти. Таким образом, виртуальная память разбивается на несколько частей, и каждая часть соответствует определенной области физической памяти.

В нашем примере, номер виртуальной страницы определяет часть виртуальной памяти, к которой относится виртуальный адрес. Например, для виртуального адреса 1230005 с номером виртуальной страницы равным 300, этот адрес будет отображаться на 300-ю страницу физической памяти. Смещение указывает на конкретный байт внутри этой страницы.

Страничное распределение виртуальной памяти позволяет эффективно управлять памятью, позволяя процессу использовать большие адресные пространства, но загружать в память только те страницы, которые действительно необходимы для выполнения задачи.