Разбор задач для подготовки к OS Рубежке.

Автор: Кривоносов Егор aka RedGry

Ссылки с разбором некоторых задач:

Издание 9 (краткая версия)
Издание 6
Издание 5
Файл Никиты Федорова
Сама книжка с задачами
Если задачи нет, то переводи ее на английский и вбивай именно в Google поиск!!!

Задача 1.4

Условие

Рассмотрим гипотетический микропроцессор, генерирующий 16-битовые адреса (предположим, например, что счетчик команд и адресные регистры имеют размер 16 бит) и обладающий 16-битовой шиной данных.

A. Какое максимальное адресное пространство памяти может быть непосредственно доступно этому процессору, если он соединен с "16-битовой памятью"?
B. Какое максимальное адресное пространство может быть непосредственно доступно этому процессору, если он соединен с "8-битовой памятью"?
C. Какие особенности архитектуры позволят этому микропроцессору получить доступ к отдельному "пространству ввода-вывода"?
D. Сколько портов ввода-вывода способен поддерживать этот микропроцессор, если в командах ввода и вывода задаются 8-битовые номера портов? Сколько портов ввода-вывода он может поддерживать с 16-битовыми портами?

Решение

A. Максимальное адресное пространство памяти: 2^16 = 64 КБайт
B. Максимальное адресное пространство памяти: 2^16 = 64 КБайт
Следовательно, в A и B микропроцессор должен получить доступ к 64 Кбайт, но разница между ними заключается в том, что доступ к 8-разрядной памяти будет передавать 8 бит, а доступ к 16-разрядной памяти может передавать 8 бит или 16 бит слово.
C. Отдельные инструкции ввода-вывода необходимы, потому что во время их выполнения будут генерироваться отдельные собственные сигналы ввода-вывода. Эти сигналы будут отличаться от сигналов памяти, которые генерируются во время

выполнения инструкции по запоминанию. Следовательно, для передачи сигналов ввода/вывода потребуется еще одна инструкция.

D. 2^8 = 256 (8 бит входных портов) и 2^8 = 256 (8 бит выходных портов). Для 16 и 8 битовых портов.
   Таким образом, размер порта ввода-вывода не изменит количество портов ввода-вывода, поскольку количество портов ввода-вывода зависит от количества битов, которое используется для представления номера порта ввода-вывода (равно 8 битам в обоих случаях).

Задача 1.5

Условие

Рассмотрим 32-битовый микропроцессор с 16-битовой внешней шиной данных, которая управляется тактовым генератором с тактовой частотой 8 МГц. Пусть цикл шины этого микропроцессора по длительности равен четырем циклам тактового генератора.

A. Какую максимальную скорость передачи данных может поддерживать этот процессор?
B. Что будет лучше для повышения производительности: сменить его внешнюю шину данных на 32-битовую или удвоить частоту сигнала тактового генератора, поступающего на микропроцессор?

Указание: определите количество байтов, которое может быть передано при каждом цикле шины

Решение

A. Поскольку минимальная продолжительность цикла шины = 4 циклам (тактам), а частота шины = 8 МГц. Тогда максимальная скорость цикла шины: 8 / 4 = 2 M/s. Данные, передаваемые за цикл шины: 16-бит = 2 байт. Значит скорость передачи данных, которую поддерживает это процессор: 2 M/s * 2 = 4 Мбайт/сек
   ОТВЕТ: $\frac{8\ \text{Мгц} \times 2\ \text{байта}}{4\ \text{циклов}} = 4$ Мбайт / сек

$$\frac{\text{Speed (MHz) x Width (bytes)}}{\text{Clock cycles per transfer}}$$

example 1-One wait state

$$\frac{\text{400MHz x 4 Bytes}}{\text{3 Cycles}}$$

533MB/s

B. Для повышения его производительности:
    a. Удвоение частоты может означать внедрение новой технологии изготовления микросхем (при условии, что каждая инструкция будет иметь одинаковое количество тактов);
    b. Удвоение внешней шины данных означает более широкие (возможно, никогда) встроенные драйверы/защелки шины данных и модификации логики управления шиной.

Поэтому в первой ситуации скорость работы микросхем памяти нужно будет удвоить, чтобы не замедлять работу микропроцессора. Что касается второй ситуации, длина слова в памяти удвоится, чтобы иметь возможность отправлять/получать 32-бит данные.

Задача 1.8

Условие

Контроллер DMA передает символы из внешнего устройства в основную память со скоростью 9600 бит в секунду. Процессор может выбирать команды со скоростью 1 млн команд в секунду.

A. **Насколько процессор замедлит свою работу из-за работы DMA?**

Решение

A. Учитывая, что процессор извлекает инструкции со скоростью 1 млн команд в секунду, то есть 1 инструкцию за 1 микросекунду. Предположим, что тактовый цикл также составляет 1 микросекунду.

   Теперь DMA передает данные со скоростью $\frac{9600}{8} = 1200\ \text{байт/с}$= 1200 символов в секунду. Значит 1 байт DMA будет передавать за $\frac{1}{1200} = 0.0008333\ldots\ \text{сек}$=> 833 микросекунды. Следовательно, контроллеру нужен доступ каждый 833 цикл. Он будет замедлять процессор на$\frac{1}{833} * 100 = 0,12004\ \%$

   Другое решение:

   Значит, процессор будет замедляться на: $\frac{1200}{1000000-1200} \times 100 = 0.12\ \%$

   Или $\frac{1200}{1000000-1200} = \frac{1200}{998800} = 0.0012014417\ldots\ \text{сек} = 1.2\ \text{миллисекунд}$

Задача 2.1

Условие

Предположим, у нас есть многозадачный компьютер, в котором каждое задание имеет идентичные характеристики. В течение цикла вычисления одного задания T половину времени занимает ввод-вывод, а вторую половину работа процессора. Для выполнения каждого задания требуется N-циклов. Допустим, что для планирования используется простой алгоритм циклического обслуживания и что ввод-вывод может выполняться одновременно с работой процессора. Определите значения следующих величин:

- Реальное время, затрачиваемое на выполнение задания.
- Среднее количество заданий, которые выполняются в течение одного цикла T.
- Доля времени, в течение которого процессор активен (не находится в режиме ожидания).

Вычислите эти значения для одного, двух и четырех одновременно выполняющихся заданий, считая, что время цикла T распределяется одним из следующих способов.

A. В течение первой половины периода выполняется ввод-вывод, а в течение второй - работа процессора.
B. В течение первой и четвертой четвертей выполняется ввод-вывод, а в течение второй и третьей - работа процессора

Решение

Ответы одинаковы для (A) и (B). Предположим, что, хотя операции процессора не могут пересекаться, а операции Ввода-Вывода могут.

2.1 The answers are the same for **(a)** and **(b)**. Assume that although processor operations cannot overlap, I/O operations can.

Number of jobs	TAT	Throughput	Processor utilization
1	NT	1/N	50%
2	NT	2/N	100%
4	(2N – 1)T	4/(2N – 1)	100%

Задача 3.2

Условие

Предположим, что в момент времени 5 не используются никакие системные ресурсы, за исключением процессора и памяти. Теперь рассмотрим следующие события.

В момент 5: P1 выполняет команду чтения с дискового устройства 3
В момент 15: Истекает квант времени P5
В момент 18: P7 выполняет команду записи на дисковое устройство 3
В момент 20: P3 выполняет команду чтения с дискового устройства 2
В момент 24: P5 выполняет команду записи на дисковое устройство 3
В момент 28: Выполняется выгрузка процесса P5 на диск
В момент 33: Прерывание от дискового устройства 2: чтение P3 завершено
В момент 36: Прерывание от дискового устройства 3: чтение P1 завершено
В момент 38: Процесс P8 завершается
В момент 40: Прерывание от дискового устройства 3: запись P5 завершена
В момент 44: Загрузка процесса P5 с диска
В момент 48: Прерывание от дискового устройства 3: запись P7 завершена

Для каждого из моментов времени 22, 37 и 47 укажите состояние, в котором находится каждый процесс. **Если процесс блокирован, укажите событие, которое к этому привело.**

Решение

В Момент 22:
P1 в Blocked state - Blocked для I/O
P3 в Blocked state - Blocked для I/O
P5 в Running state устройства 3
P7 в Blocked state - Blocked для I/O
P8 в Running state устройства 1

В Момент 37:
P1 в Running state устройства 3
P3 в Running state устройства 2
P5 в Blocked или Suspended state из-за прерывания со стороны P1
P7 в Blocked для I/O
P8 в Running state устройства 1

В Момент 47:
P1 в Running state устройства 3
P3 в Running state устройства 2
P5 в Ready state
P7 в Blocked для I/O
P8 в Exit state

Задача 7.2

Условие

Рассмотрим схему фиксированного распределения с разделами равного размера, равного 2^16 байт, и общим количеством основной памяти 2^24 байт. Поддерживается таблица процессов, включающая указатель на раздел для каждого резидентного процесса.
Сколько битов требуется для этого указателя?

Решение

Количество разделов равно количеству байтов основной памяти, деленному на количество байтов в каждом разделе: 2^24/2^16 = 2^8. Для идентификации одного из разделов 2^8 требуется **восемь битов**.

Задача 7.12

Условие

Рассмотрим простую страничную систему со следующими параметрами: 2^32 байт физической памяти; размер страниц - 2^10 байт; 2^16 страниц логического адресного пространства.

A. Сколько битов в логическом адресе?
B. Сколько байт в кадре?
C. Сколько битов физического адреса определяет кадр?
D. Сколько записей в таблице страниц?
E. Сколько битов в каждой записи таблицы страниц? Предполагаем, что каждая запись таблицы страниц содержит бит корректности страницы.

Решение

A. Это биты адреса страницы плюс количество битов страницы. Верхняя часть адреса - это номер страницы (16 бит), а нижняя часть - это смещение внутри этого адреса (10 бит), поэтому общий размер адреса составляет **26 бит** ($2^{16} \times 2^{10} = 2^{26}$).
B. Кадр - это место, где страница может быть отображена в памяти, поэтому фрейм должен быть того же размера, что и страница **(**2^{10}байт**)**
C. Итак, у вас есть 32 бита физического адреса, а размер кадра составляет 2^10 , так что остается **22 бита** (32–10) для базового адреса кадра.
D. Таблица страниц - это полный список страниц, отображенных или не отображенных - так что в таблице страниц **2^16 записей**, поскольку есть 2^16 страниц.

E. Если каждая страница соответствует записи в таблице страниц, и эта таблица представляет собой список адресов, по которым страницы отображаются в физической памяти, то каждый адрес в таблице должен иметь размер (22 бита) плюс один бит для бита корректности, поэтому **23 бита**.

Задача 7.14

Условие

Рассмотрим простую систему сегментации, при которой используется следующая таблица сегментов:

Начальный адрес	**Длина (байты)**
660	248
1752	422
222	198
996	604

Для каждого из следующих логических адресов определите физический адрес или укажите, что заданный адрес ошибочен.

A. 0 | 198
B. 2 | 156
C. 1 | 530
D. 3 | 444
E. 0 | 222

Решение

Во-первых, нужно проверить, нет ли (смещение < предел сегмента) проблемы. Если это так, добавляем смещение + базовый адрес для физического адреса. Если это не так, произойдет ошибка сегмента.

A. 198 < 248 - TRUE => 660 + 198 = **858**
B. 156 < 198 - TRUE => 222 + 156 = **378**
C. 530 < 422 - FALSE => **заданный адрес ошибочен**
D. 444 < 604 - TRUE => 996 + 444 = **1440**
E. 222 < 248 - TRUE => 660 + 222 = **882**

Задача 8.1

Условие

Предположим, что таблица страниц текущего процесса выглядит так, как показано ниже. Все числа в таблице - десятичные, вся нумерация начинается с нуля, а все адреса представляют собой адреса отдельных байтов памяти. Размер страницы равен 1024 байтам.

Номер виртуальной страницы	**Бит присутствия в памяти**	**Бит обращений**	**Бит модификации**	**Номер кадра**
0	1	1	0	4
1	1	1	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

A. Опишите, как именно виртуальный адрес транслируется в физический адрес основной памяти.
B. Какой физический адрес (если таковой имеется) соответствует каждому из приведенных виртуальных адресов? (Вы не должны пытаться обработать прерывание из-за отсутствия страницы) .
    - 1052
    - 2221
    - 5499

Решение

A. Разделяем двоичный адрес на номер виртуальной страницы (НВС) и смещение. Затем, используя НВС в качестве индекса в таблице страниц. Извлекаем номер кадра страницы из записи таблицы страниц. Объединяем номер кадра страницы со смещением, чтобы получить адрес физической памяти.
B. Решение:
    a. 1052 = 1 * 1024 + 28 сопоставляется с НВС (1), где номер кадра 7 => (**7 * 1024 + 28 = 7196**)
    b. 2221 = 2 * 1024 + 173 сопоставляется с НВС (2) => **ошибка страницы**

c. 5499 = 5 * 1024 + 379 сопоставляется с НВС (5), где номер кадра 0 => (**0 * 1024 + 379 = 379**)

Задача 8.4

Условие

Рассмотрим последовательность обращений к страницам
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2.
Изобразите диаграмму, подобную показанной на рис. 8.14 и демонстрирующую распределение кадров для стратегий.

A. FIFO ("первым вошел первым вышел").
B. LRU (последний использовавшийся).
C. Часовой.
D. Оптимальный (в предположении, что последовательность обращений продолжается как 1, 2, 0, 1, 7, 0, 1).
E. Перечислите общее количество ошибок страниц и частоту промахов для каждой стратегии.

Подсчитайте количество ошибок страницы, происшедших после того, как все кадры были инициализированы.

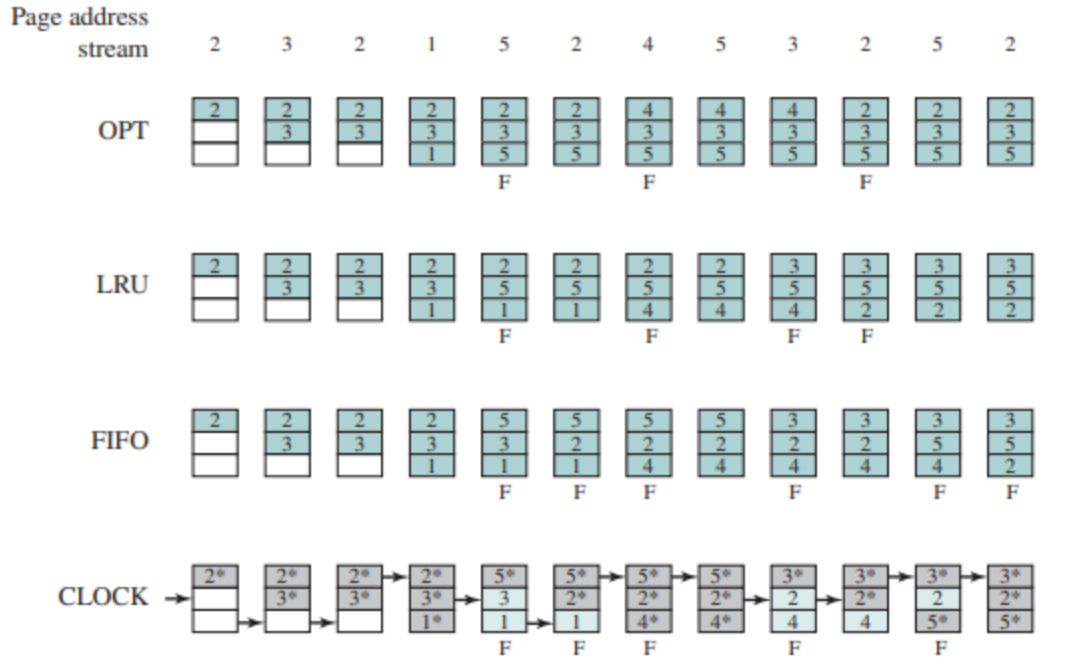


Рис. 8.14. Поведение четырех алгоритмов замещения страниц

Решение

A.

FIFO	7	7	7	2	2	2	2	4	4	4	0	0	0
		0	0	0	0	3	3	3	2	2	2	2	2
			1	1	1	1	0	0	0	3	3	3	3
				F		F	F	F	F	F	F		

B.

LRU	7	7	7	2	2	2	2	4	4	4	0	0	0
		0	0	0	0	0	0	0	0	3	3	3	3
			1	1	1	3	3	3	2	2	2	2	2
				F		F		F	F	F	F		

C.

Часовой	7*	7*	7*	2*	2*	2*	2*	4*	4*	4*	4	3*	3*
		0*	0*	0	0*	0	0*	0	2*	2*	2	2	2*
			1*	1	1	3*	3*	3	3	3*	0*	0*	0*
				F		F		F	F		F	F	

D.

OPT	7	7	7	2	2	2	2	2	2	2	2	2	2
		0	0	0	0	0	0	4	4	4	0	0	0
			1	1	1	3	3	3	3	3	3	3	3
				F		F		F			F		

E. FIFO: промахов - 10
LRU: промахов - 9
CLOCK: промахов - 9
OPT: промахов - 7

Page references: 7,0,1,2,0,3,0,4,2,3,0,3,2

FIFO

7	7	7	2	2	2	2	4	4	4	0	0	0
	0	0	0	0	3	3	3	2	2	2	2	2
		1	1	1	1	0	0	0	3	3	3	3
			F		F	F	F	F	F	F		

LRU

7	7	7	2	2	2	2	4	4	4	0	0	0
	0	0	0	0	0	0	0	0	3	3	3	3
		1	1	1	3	3	3	2	2	2	2	2
			F		F		F	F	F	F		

CLOCK (a gray frame represents the pointer)

7^1	7^1	7^1	2^1	2^1	2^1	2^1	4^1	4^1	4^1	4^0	3^1	3^1
	0^1	0^1	0^0	0^1	0^0	0^1	0^0	2^1	2^1	2^0	2^0	2^1
		1^1	1^0	1^0	3^1	3^1	3^0	3^0	3^1	0^1	0^1	0^1
			F		F		F	F		F	F	

Задача 8.6

Условие

Процесс содержит восемь виртуальных страниц на диске, и ему выделено четыре фиксированных кадра в основной памяти. Далее выполняются обращения к следующим страницам:
1, 0, 2, 2, 1, 7, 6, 7, 0, 1, 2, 0, 3, 0, 4, 5, 1, 5, 2, 4, 5, 6, 7, 6, 7, 2, 4, 2, 7, 3, 3, 2, 3.

A. Укажите последовательность размещения страниц в кадрах при использовании алгоритма замещения наиболее долго не использовавшейся страницы. Вычислите результативность обращения к основной памяти (считаем, что изначально все кадры пусты).
B. Выполните то же задание для алгоритма "первым вошел - первым вышел".
C. Сравните результативности обращения к основной памяти, вычисленные в первых двух заданиях, и прокомментируйте эффективность использования указанных алгоритмов применительно к данной последовательности обращений.

Решение

A. LRU: вероятность попадания 16/33
B. FIFO: вероятность попадания 16/33
C. Эти два алгоритма одинаково эффективны для этой конкретной трассировки страниц.

Задача 8.10

Условие

Предположим, что размер страницы занимает 4 Кбайт и что запись таблицы страниц занимает 4 байт. Сколько уровней таблиц страниц потребуется для отображения 64-битового адресного пространства, если таблица верхнего уровня занимает одну страницу?

Решение

Поскольку каждая запись таблицы страниц составляет 4 байта, а каждая страница содержит 4 Кбайта, то одностраничная таблица страниц будет указывать на 1024 = 2^10 страниц, в общей сложности 2^10 × 2^12 = 2^22 байта. Однако адресное пространство составляет 2^64 байта. Добавляя второй слой таблиц страниц, верхняя таблица страниц будет указывать на таблицы на 2^10 страниц, адресуя в общей сложности 2^32 байта. Продолжая этот процесс, мы видим, что 5 уровней не адресуют все 64-разрядное адресное пространство, поэтому требуется 6-й уровень. Но требуется только 2 бита 6-го

уровня, а не все 10 бит. Таким образом, вместо того, чтобы требовать, чтобы ваши виртуальные адреса были длиной 72 бита, вы могли бы замаскироваться и игнорировать все, кроме 2 младших разрядов 6-го уровня. Это даст вам 64-разрядный адрес. Тогда в вашей таблице страниц верхнего уровня будет всего 4 записи. Еще один вариант - пересмотреть критерии, по которым таблица страниц верхнего уровня помещается на одной физической странице, и вместо этого разместить ее на 4 страницах. Это позволило бы сэкономить физическую страницу, а это не так уж много.

Depth	Address Space
1	2^{22} bytes
2	2^{32} bytes
3	2^{42} bytes
4	2^{52} bytes
5	2^{62} bytes
6	2^{72} bytes ($> 2^{64}$ bytes)

ОТВЕТ: 6

Задача 8.17

Условие

Предположим, что задание разделено на четыре сегмента одинакового размера и что для каждого сегмента система строит таблицу дескрипторов страниц с восемью записями. Таким образом, описанная система представляет собой комбинацию сегментации и страничной организации. Предположим также, что размер страницы равен 2 Кбайт.

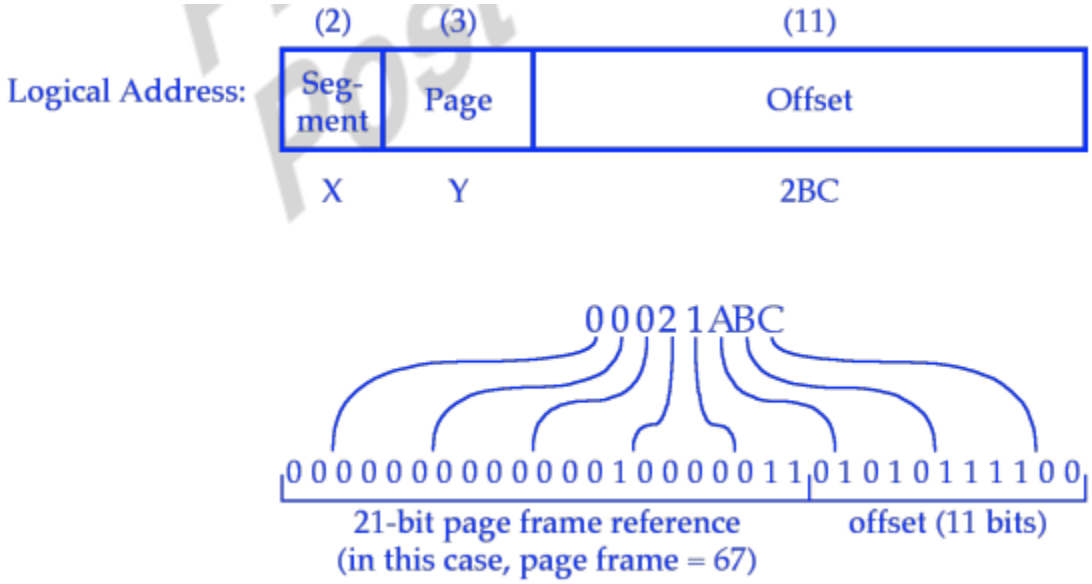
A. Чему равен максимальный объем каждого сегмента?
B. Каково максимальное логическое адресное пространство одного задания?
C. Предположим, что рассматриваемое задание обратилось к ячейке памяти с физическим адресом 0002 1ABC. Каков формат генерируемого для этого логического адреса? Каково максимально возможное физическое адресное пространство в этой системе?

Решение

A. (8 записей в таблице страниц) x 2 Кбайт = 16 Кбайт.
B. (16 Кбайт) x 4 сегмента на задачу = 64 Кбайт.
C. Мы знаем, что смещение составляет 11 бит, так как размер страницы составляет 2 КБ. Таблица страниц для каждого сегмента содержит восемь записей, поэтому ей требуется 3 бита. Это оставляет 2 бита для номера сегмента. Таким образом, формат адреса составляет 2 бита для номера сегмента, 3 бита для номера страницы и 11 бит для смещения. Общий физический адрес имеет ширину 32 бита,

поэтому номер кадра должен иметь ширину 21 бит. Таким образом, 0002 1ABC представлен в двоичном виде как:

Кадр	Смещение
0000 0000 0000 0010 0001 1	010 1011 1100



Максимальное физическое адресное пространство составляет 2^32 = 4 ГБайт.

Обратите внимание, что виртуальное адресное пространство имеет ширину всего 16 бит: 11 для смещения, 3 для номера страницы и 2 для номера сегмента. Это означает, что процесс не может логически охватить все пространство объемом 4 ГБайт... он ограничен всего 64 Кбайт.

Задача 9.16

Условие

Пять пакетных заданий, от A до E, поступают в вычислительный центр одновременно. Их ожидаемое время работы - 15, 9, 3, 6 и 12 минут соответственно. Их приоритеты, определенные при передаче заданий, равны соответственно 6, 3, 7, 9 и 4, причем меньшее значение означает более высокий приоритет.
Для каждого из перечисленных ниже алгоритмов определите время оборота каждого процесса и среднее время оборота всех процессов. Накладные расходы, связанные с переключением процессов, не учитываются. Поясните, как вы пришли к данному ответу. В трех последних случаях предполагается, что в определенный момент времени работает только один процесс, вытеснения не происходит и все задания ориентированы на вычисления.

A. Круговое планирование с размером кванта, равным 1 минуте.

B. Планирование с учетом приоритетов.
C. FCFS при запуске процессов в следующем порядке: 15, 9, 3, 6 и 12.
D. Первым выполняется самое короткое задание.

Решение

A.

a. Sequence with which processes will get 1 min of processor time:

1	2	3	4	5	Elapsed time
A	B	C	D	E	5
A	B	C	D	E	10
A	B	C	D	E	15
A	B		D	E	19
A	B		D	E	23
A	B		D	E	27
A	B			E	30
A	B			E	33
A	B			E	36
A				E	38
A				E	40
A				E	42
A					43
A					44
A					45

The turnaround time for each process:
A = 45 min, B = 35 min, C = 13 min, D = 26 min, E = 42 min
The average turnaround time is = (45+35+13+26+42) / 5 = 32.2 min

B.

b.

Priority	Job	Turnaround Time
3	B	9
4	E	9 + 12 = 21
6	A	21 + 15 = 36
7	C	36 + 3 = 39
9	D	39 + 6 = 45

The average turnaround time is: (9+21+36+39+45) / 5 = 30 min

C.

c.

Job	Turnaround Time
A	15
B	15 + 9 = 24
C	24 + 3 = 27
D	27 + 6 = 33
E	33 + 12 = 45

The average turnaround time is: (15+24+27+33+45) / 5 = 28.8 min

D.

d.

Running Time	Job	Turnaround Time
3	C	3
6	D	3 + 6 = 9
9	B	9 + 9 = 18
12	E	18 + 12 = 30
15	A	30 + 15 = 45

The average turnaround time is: (3+9+18+30+45) / 5 = 21 min

Задача 11.7

Условие

Рассчитайте количество дискового пространства (в секторах, дорожках и поверхностях), необходимого для хранения 300 000 120-байтных логических записей, если диск разбит на секторы размером 512 байт, с 96 секторами на дорожке, 110 дорожками на поверхности и 8 используемыми поверхностями. Служебные записи о файле во внимание не принимайте; считайте также, что запись не может быть разбита и размещена на двух секторах.

Решение

Рассмотрите следующую информацию:

- Количество байт/сектор=512
- Размер каждой логической записи=120 байт
- Количество записей =300 000

Таким образом, количество логических записей, которые может содержать каждый сектор:
$\frac{512}{120} \simeq 4$

- Количество записей в секторе = 4

Дисковое пространство в секторах для хранения 300 000 записей:
$\frac{300000}{4} = 75000$**секторов.**

- Количество секторов в 300 000 записях = 75 000
- Количество секторов на дорожку = 96

Дисковое пространство в дорожках для хранения 300 000 записей:
$\frac{75000}{96} \simeq 782$**дорожки.**

- Количество дорожек в 300 000 записях = 782
- Количество дорожек на поверхности = 110

Дисковое пространство в поверхностях для хранения 300 000 записей:
$\frac{782}{110} \simeq 8$**поверхностей**

Задача 11.12

Условие

Рассмотрим RAID-массив из четырех 200-гигабайтных дисков. Какова доступная для хранения данных емкость в случае использования каждого из уровней RAID - 0,1, 3, 4, 5, 6?

Решение

Количество дисков с доступной для хранения данных:

- RAID-0: N
- RAID-1: N / 2
- RAID-3: N - 1
- RAID-4: N - 1
- RAID-5: N - 1
- RAID-6: N - 2

Где N - это исходное количество дисков.

RAID-0: 200 Гбайт * 4 = 800 Гбайт
RAID-1: 200 Гбайт * 2 = 400 Гбайт
RAID-3: 200 Гбайт * 3 = 600 Гбайт
RAID-4: 200 Гбайт * 3 = 600 Гбайт
RAID-5: 200 Гбайт * 3 = 600 Гбайт

RAID-6: 200 Гбайт * 2 = 400 Гбайт

Сравнение RAID-систем

RAID	Минимум дисков	Потребность в дисках	Отказо-устойчивость	Скорость передачи данных	Интенсивность обработки запросов	Практическое использование
0	2	N	нет	< RAID 3	очень высокая до N x 1 диск	Графика, видео
1	2	2N	1 диск	R > 1 диск W = 1 диск	до 2 x 1 диск W = 1 диск	малые файл-серверы
2	7	2N >X>N+1	1 диск	~ RAID 3	Низкая	мейнфреймы
3	3	N+1	1 диск	низкая	Низкая	Графика, видео
4	3	N+1	1 диск	R < RAID 3 W < RAID 5	R = RAID 0 W << 1 диск	файл-серверы
5	3	N+1	1 диск	R < RAID 4 W < RAID 3	R = RAID 0 W < 1 диск	серверы баз данных (обработка транзакций)
6	4	N+2	2 диска	низкая	R > 1 диск W < RAID 4	используется крайне редко

Задача 12.7

Условие

Игнорируя накладные расходы на дескрипторы каталогов и файлов, рассмотрите файловую систему, в которой файлы хранятся в блоках размером 16 Кбайт. Для каждого из следующих размеров файлов рассчитайте процент потерянного файлового пространства из-за неполного заполнения последнего блока: 41 600, 640 000, 4 064 000 байт.

Решение

A. Размер файла 41 600 байт
   Количество блоков: $\frac{41\,600}{16 \times 1024} \approx 3$(округляем в большую сторону до целого числа)
   Общая вместимость блоков: $3 \times 16\,\text{Кбайт} = 48\,\text{Кбайт} = 49\,152\,\text{байт}$

Потрачено в пустую: $49\,152 \; - \; 41\,600 \; = \; 7\,552$ байт

В процентах: $\frac{7\,552}{49\,152} \times 100 \approx 15.36\%$

B. Размер файла 640 000 байт

Количество блоков: $\frac{640\,000}{16 \times 1024} \approx 40$(округляем в большую сторону до целого числа)

Общая вместимость блоков: $40 \; \times 16$ Кбайт $\; = \; 640$ Кбайт $\; = \; 655\,360$ байт

Потрачено в пустую: $655\,360 \; - \; 640\,000 \; = \; 15\,360$ байт

В процентах: $\frac{15\,360}{655\,360} \times 100 \approx 2.34\%$

C. Размер файла 4 064 000 байт

Количество блоков: $\frac{4\,064\,000}{16 \times 1024} \approx 249$(округляем в большую сторону до целого числа)

Общая вместимость блоков: $249 \; \times 16$ Кбайт $\; = \; 3\,984$ Кбайт $\; = \; 4\,079\,616$ байт

Потрачено в пустую: $4\,079\,616 \; - \; 4\,064\,000 \; = \; 15\,616$ байт

В процентах: $\frac{15\,616}{4\,079\,616} \times 100 \approx 0.38\%$

Задача 12.12

Условие

В UNIX System V длина блока составляет 1 Кбайт, а каждый блок может содержать до 256 адресов блоков. Каков максимальный размер файла при использовании схемы индексных узлов?

Решение

Unix использует многоуровневый механизм индексирования для размещения файлов на диске. Адреса первых 10 блоков данных + 3 индексных блока (первый, второй и третий уровни индексации).
Согласно приведенным выше параметрам, максимальный размер файла составляет чуть более 16 Гбайт.

Первые десять адресов указывают на первые 12 блоков данных файла = 12 * 1 Кбайт
1 уровень = 256^1 * 1 Кбайт
2 уровень = 256^2 * 1 Кбайт
3 уровень = 256^3 * 1 Кбайт

А если точнее: $256^3 + 2 \times 256^2 + 256^1 + 12 = 16843020$ Кбайт
Подробнее можно посмотреть на рисунке 12.15 в задаче 12.13

Задача 12.13

Условие

Рассмотрим организацию файлов UNIX, представленную на рис. 12.15. Пусть в каждом узле содержится 12 прямых указателей блоков, а также одинарный, двойной и тройной указатели. Далее положим, что размер системного блока и размер дискового сектора равны 8 Кбайт. Допустим, что размер указателя дискового блока – 32 бита (8 бит для указания физического диска и 24 бита для указания физического блока).

A. Какой максимальный размер файла, поддерживаемый в этой системе?
B. Какой максимальный размер раздела файловой системы, поддерживаемого в этой системе?
C. Предположим, что в основной памяти не содержится ничего, кроме индексного узла. Сколько обращений к диску потребуется для доступа к байту в позиции 13 423 956?

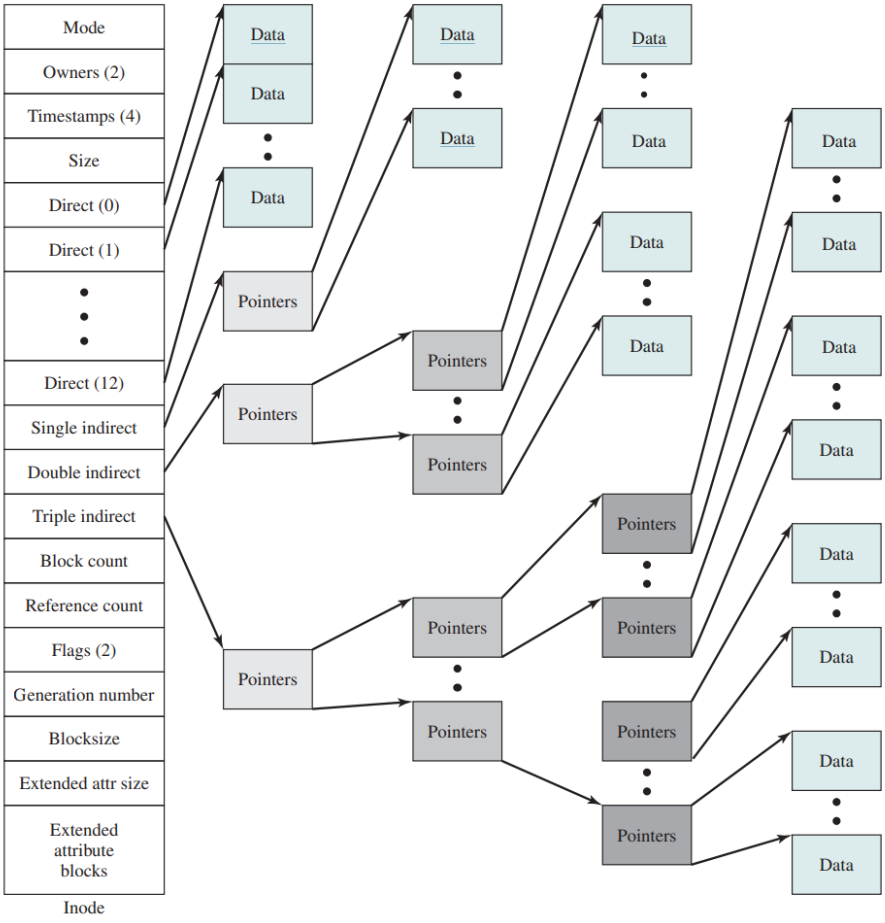


Figure 12.15 Structure of FreeBSD Inode and File

Решение

A. Максимальный размер файла можно определить путем вычисления пространства, на которое могут ссылаться все различные типы указателей блоков. Во-первых, нам нужно рассчитать количество указателей, которые может содержать косвенный блок:

$N = 8 \times 1024 \times \frac{8}{32} = 2048$ адресов в блоке

Тогда,
Прямой уровень = 12 * 8 Кбайт
1 уровень = 2048^1 * 8 Кбайт
2 уровень = 2048^2 * 8 Кбайт
3 уровень = 2048^3 * 8 Кбайт

Максимальный размер файла: 68753047648 Кбайт ≈ 64 Тбайт

B. Максимальный раздел файловой системы по существу равен размеру диска. Поскольку мы используем 24 бита для адресации блоков на каждом диске, максимальный размер диска составляет **128 ГБ**.

C. Указанный адрес находится в диапазоне 13 Мбайт. 12 прямых указателей охватывают 96 Кбайт, поэтому адрес находится не там. Однонаправленный косвенный указатель охватывает следующие 16 Мбайт файла, поэтому адрес находится в блоке, на который ссылается однонаправленный косвенный указатель. **Это означает, что нам потребуется два доступа к диску, один для косвенного блока, а другой для блока, содержащего данные.**

Задача 1.13 (Доп)

Условие

В компьютере есть кеш, основная память и диск, выступающий в роли виртуальной памяти. Если запрашиваемое слово находится в кэше, для доступа к нему требуется 20 нс. Если запрашиваемое слово находится не в кеше, а в основной памяти, для его загрузки в кэш требуется 60 нс (сюда входит время, которое требуется для первоначальной проверки кеша). После этого происходит новый запрос. Если слова нет в оперативной памяти, чтобы получить его с диска, необходимо затратить 12 мс, а затем еще 60 нс, чтобы скопировать его в кеш; после этого происходит новый запрос. Результативность поиска в кеше равна – 0.9, а результативность поиска в основной памяти 0.6. **Найти среднее время, которое требуется для получения доступа к слову в данной системе.**

Решение

Location of referenced word	Probability	Total time for access in ns
In cache	0.9	20
Not in cache, but in main memory	(0.1)(0.6) = 0.06	60 + 20 = 80
Not in cache or main memory	(0.1)(0.4) = 0.04	12ms + 60 + 20 = 12,000,080

So the average access time would be:

Avg = (0.9)(20) + (0.06)(80) + (0.04)(12000080) = 480026 ns

Question shows a hierarchical memory system

$$T_{avg} = hit_{cache} \times time_{cache}$$
$$+ (1 - hit_{cache})\,(hit_{memory})\,[time_{cache} + time_{memory}]$$
$$+ (1 - hit_{cache})\,(1 - hit_{memory})\,[time_{cache} + time_{memory} + time_{disk}]$$

$$= 0.9 \times 20$$
$$+ 0.1 \times 0.6 \times 80$$
$$+ 0.1 \times .4 \times 12000080$$

We can also use the formula

$$T_{avg} = time_{cache}$$
$$+ (1 - hit_{cache})\,[time_{memory}]$$
$$+ (1 - hit_{cache})\,(1 - hit_{memory})\,[time_{disk}]$$

$$= 20 + 0.1 \times 60 + 0.1 \times 0.4 \times 12000000$$