

Семестровая работа

1. Сколько времени занимает считывание диска с 800 цилиндрами, каждый из которых содержит 5 дорожек по 32 сектора? Сначала считываются все сектора дорожки 0, начиная с сектора 0, затем все сектора дорожки 1, начиная с сектора 0 и т.д. Оборот совершается за 20мс, поиск между соседними цилиндрами занимает 10мс, а в случае расположения считываемых данных в разных частях диска — до 50мс. Переход от одной дорожки цилиндра к другой происходит мгновенно.

Решение:

5 дорожек * 20мс = 100 мс — время считывания одного цилиндра. 100мс * 800 = 80с = t время считывания всех цилиндров

считывание проходит между соседними цилиндрами $t_{min} = 7,99с$
if { считывание проходит по цилиндрам с максимальным разбросом, то $799 * 10мс = t_{min} = 7,99с$
 $799 * 50мс = t_{max} = 39,95с$

Время считывания диска будет лежать в пределах от t + t_{min} до t + t_{max}. Следовательно, считывание диска занимает от 87,99с до 119,95с

2. Представьте, что вы записываете часть операционной системы, отвечающую за управление диском. Логически вы представляете себе диск как последовательность блоков от 0 на внутренней стороне до какого-либо максимума снаружи. Когда создаются файлы, вам приходится размещать свободные сектора. Вы можете двигаться от наружного края внутрь или наоборот. Имеет ли значение, какую стратегию выбрать? Поясните свой ответ.

Решение:

Мне кажется двигаться от наружного края к внутреннему выгоднее, потому что идите нахуй

3. Система адресации LBA использует 24 бита для обращения к сектору. Каков максимальный объем диска, с которым она может работать?

Решение:

$2^{24} = 16777216$ — кол — во секторов. Один сектор как правило содержит 512 Мбайт. Следовательно, максимальный объем диска составляет кол-во секторов * объем одного сектора = 8589934592 Мбайт = 8388608 Гбайт = 8192 Тбайт = 8 Пбайт

4. Чтобы вместить фильм длительностью 133 минуты на односторонний DVD с одним слоем, требуется небольшая компрессия.

Вычислите, насколько нужно сжать фильм. Предполагается, что для записи дорожки изображения нужно 3,5 Гбайт, разрешающая способность изображения 720x480 пикселей с 24-битным цветом и в секунду меняется 30 кадров.

Решение:

Кол-во пикселей (разрешение) $N = 720 * 480 = 345600$

Размер одного кадра $R = N * \text{размер (вес) одного пикселя} = 345600 * 24 \text{бит} = 8294400 \text{бит} = 1036800 \text{байт} = 1012,5 \text{Кбайт}$

Размер одной секунды изображения $V = R * 30 \text{кадров} = 30375 \text{Кбайт}$

Размер изображения фильма $X = V * 133 * 60с = 242392500 \text{Кбайт} \approx 231 \text{Гбайт}$

Фильм надо сжать на $231 - 3,5 = 227,5 \text{Гбайт}$, но обычно вычисляется величина отношения размеров (она более информативна).

Следовательно, фильм необходимо сжать в $231 / 3,5 = 66$ раз.

5. Предположим, что центральный процессор (CPU) содержит кэш-память первого и второго уровня со временем доступа 5нс и 10нс соответственно. Время доступа к основной памяти составляет 50нс. Если 20% от всех обращений к памяти приходится на долю кэш-памяти первого уровня, а 60% — на долю кэш-памяти второго уровня, то каково среднее время доступа?

Решение:

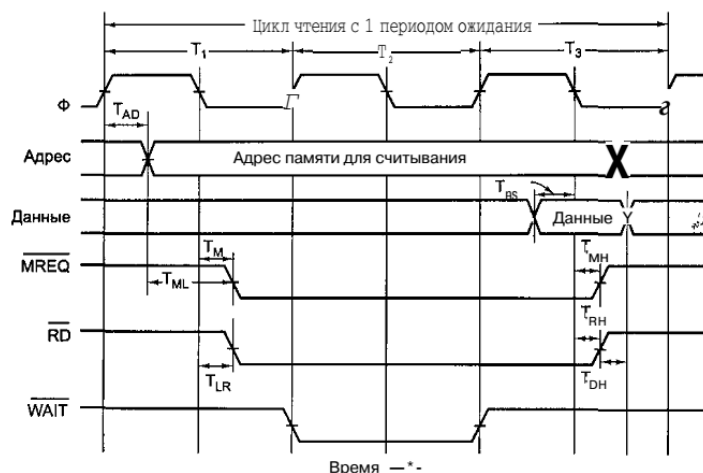
p_1 — вероятность при запросе оказаться в кэше первого уровня. $p_1 = 1$ т.к. при любом запросе мы сначала оказываемся на кэш-памяти первого уровня.

p_2 — вероятность при запросе оказаться в кэше второго уровня. $p_2 = 1 - 0.2 = 0.8$ кэш-память первого уровня обрабатывает только 20% запросов, остальные попадают в кэш-память второго уровня.

p_3 — вероятность при запросе оказаться в основной памяти. $p_3 = p_2 - 0.6 = 0.2$ если запрос был не найден в кэш-памяти то он попадает в основную память.

$\langle t \rangle = t_1 p_1 + t_2 p_2 + t_3 p_3 = 5нс * 1 + 10нс * 0.8 + 50нс * 0.2 = 23нс$

6. На рисунке приведена временная диаграмма процесса считывания на синхронной шине. Предположим, что тактовый генератор работает с частотой 40 МГц, а T_{AD} возросло до 16нс. Можно ли при этом продолжать использовать микросхемы памяти на 40нс?



7.1. Компьютер содержит двухуровневую кэш-память. Предположим, что 80%

обращений к памяти—удачные обращения в кэш-память первого уровня, 15%—в кэш-память второго уровня, а 5%—промахи кэша. Время доступа составляет 5нс, 15нс и 60нс соответственно, причем время доступа в кэш-память второго уровня и в основную память отсчитывается с того момента, как стало известно, что они нужны (например, доступ к кэш-памяти второго уровня не может начаться, пока не произойдет промах кэш-памяти первого уровня). Каково среднее время доступа?

Решение данной задачи аналогично решению задачи 5. Следовательно $\langle t \rangle = t_{p1} + t_{p2} + t_{p3} = 5\text{нс} \cdot 1 + 15\text{нс} \cdot 0.2 + 60\text{нс} \cdot 0.05 = 11\text{нс}$

7.2. Компьютер с конвейером из пяти стадий при обработке условных переходов простаивает следующие три цикла. Насколько эти простаивания снижат производительность, если 20% команд являются условными переходами? Другие причины простаиваний не учитывайте.

Решение:

8. Предположим, что компьютер вызывает до 20 команд заранее. В среднем 4 из этих команд являются условными переходами, причем вероятность правильного прогнозирования каждого из этих условных переходов равно 90%. Какова вероятность, что предварительный вызов команд на правильном пути?

Решение:

9. Компьютер для считывания информации с диска использует канал прямого доступа к памяти. Диск содержит 64 сектора по 512 байтов на дорожке. Время оборота диска 16мс. Ширина шины 16 битов. Каждая передача шины занимает 500нс. В среднем для одной команды процессора требуется два цикла шины. Насколько скорость работы процессора замедляется из-за прямого доступа к памяти?

10. Вычислите логическое выражение $(A \& B) \vee C$ для:

•A - 1101000010101101

•B - 111 1111100001111

•C = 0000000000100000.

Решение:

A	1	1	0	1	0	0	0	0	1	0	1	0	1	1	0	1
B	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
A&B	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	1
C	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
A&BvC	1	1	0	1	0	0	0	0	0	0	1	0	1	1	0	1

11. Переделайте следующие формулы из обратной польской записи (форма ПОЛИЗ – используемая для вычислений на стековой машине) в инфиксную запись:

•AB+C+D x

•AB/CD/+

•ABCDE+xx/

•ABCDEF/+G-H/x+

Решение: обламывает писать

12. Компьютер имеет 16 страниц виртуального адресного пространства и только 4 страничных кадра. Изначально память пуста. Программа обращается к виртуальным страницам в следующем порядке:0,7,2,7,5,8,9,2,4

а. Какие из обращений вызовут ошибку с алгоритмом LRU?

б. Какие из обращений вызовут ошибку с алгоритмом FIFO?

Решение:

BC – виртуальная страница

О№ - номер ошибки (fault'a)

Цифрами 1,2,3,4 обозначены номера заполненных страничных кадров

	FIFO									LRU								
BC	0	7	2	7	5	8	9	2	4	0	7	2	7	5	8	9	2	4
	1	2	3	2	4					1	2	3	2	4				
O1			3	2	4	1					2	3	2	4	1			
O2			3		4	1	2	3			2		2	4	1	3		
O3					4	1	2		3		2		2		1	3	4	
O4											2		2			3	4	1

13. Сегментированная память содержит страничные сегменты. Каждый виртуальный адрес содержит 2-битный номер сегмента, 2-битный номер страницы и 11-битное смещение внутри страницы. Основная память содержит 32 Кбайт, которые разделены на страницы по 2Кбайт. Каждый сегмент разрешается либо только читать, либо читать и выполнять, либо читать и записывать, либо читать,записывать и выполнять. Таблицы страниц с указанием на защиту приведены ниже:

Сегмент 0		Сегмент 1		Сегмент 2	Сегмент 3	
Только для чтения		Чтение/выполнение			Чтение/запись/ выполнение	
Вирту- альная страница	Странич- ный кадр	Вирту- альная страница	Странич- ный кадр		Вирту- альная страница ¹	Странич- ный кадр
0	9	0	На диске	Таблицы	0	14
1	3	1	0	страниц нет	1	1
2	На диске	2	15	в основной	2	6
3	12	3	8	памяти	3	На диске

Вычислите физический адрес для каждого из ниже перечисленных доступов к виртуальной памяти. Если происходит ошибка, скажите, какого она типа.

Доступ	Сегмент	Страница	Смещение внутри страницы
1. Вызов данных	0	1	1
2. Вызов данных	1	1	10
3. Вызов данных	3	3	2047
4. Сохранение данных	0	1	4
5. Сохранение данных	3	1	2
6. Сохранени данных	3	0	14
7. Переход	1	3	100
8. Вызов данных	0	2	50
9. Вызов данных	2	0	5
10. Переход	3	0	60

Решение:

14. 1% определенной программы отвечает за 50% времени выполнения этой программы. Сравните следующие три стратегии с точки зрения времени программирования и времени выполнения. Предположим, что для написания программы на языке C потребуется 10 0 человеко-месяцев, а программу на языке ассемблера написать в 10 раз труднее, но зато она работает в 4 раза эффективнее.

1. Вся программа написана на языке C.

2. Вся программа написана на ассемблере.

3. Программа сначала написана на C, а затем нужный 1% программы переписан на ассемблере.

Решение:

	Затраты на программирование. [у.е.]	Время выполнения. [время]
1	100	4t
2	$10 \cdot 100 = 1000$	t
3	$100 + 1000/100 = 110$	$t/2 + 4t/2 = 2.5t$