

ОСи рубежка

1.4 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.

а. Какое максимальное адресное пространство памяти может быть непосредственно доступно этому процессору, если он соединен с "16-битовой памятью"?

б. Какое максимальное адресное пространство может быть непосредственно доступно этому процессору, если он соединен с "8-битовой памятью"?

И в а) и в б) будет доступ к адресному пространству в количестве $2^{16} = 64$ Кбайта.

в. Какие особенности архитектуры позволят этому микропроцессору получить доступ к отдельному "пространству ввода-вывода"?

Нужны дополнительные инструкции ввода вывода

г. Сколько портов ввода-вывода способен поддерживать этот микропроцессор, если в командах ввода и вывода задаются 8-битные номера портов (под адресацию отведено 8 бит)? Сколько портов ввода-вывода он может поддерживать с 16-битовыми портами?

256, но вообще зависит от того, сколько бит отведено под адресацию в командах ввода вывода

1.5 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. >

Какую максимальную скорость передачи данных может поддерживать этот процессор?

Скорость данных: 4 Мбайта/с

Пояснение: по формуле мы можем вычислить время тактового цикла, которое будет равно $1: 8\text{Мг} = 125\text{ нс}$. Значит, Цикл шины будет равен $125 \cdot 4 = 500\text{ нс}$. Так как у нас 16 битная шина данных, то каждый цикл у нас передается 2 байта. То есть, каждые 500 нс передаются 2 байта. Следовательно 4 Мбайта/с.

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. ??????

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

Ответ: 0.12%

Пояснение: Нашему процессору нужен доступ к памяти каждую микросекунду, так как каждую секунду он передает 1 млн команд. Также каждую секунду контроллер DMA будет передавать 9600 бит = 1200 байт информации, значит, один байт информации он будет передавать за $(1 / 1200 \text{ байт}) \Rightarrow 833 \text{ мкс}$. Следовательно, контроллеру нужен доступ каждый 833 цикл. Он будет замедлять процессор на $(\text{миллион команд в секунду разделить на мкс}) \Rightarrow 1:833 * 100\% = 0.12\%$.

2.1 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 одного задания.

1: NT

2: NT

4: $(2N-1)T$

- Среднее количество заданий, которые выполняются в течение одного цикла T .

1: $1/N$

2: $2/N$

4: $4/(2N-1)$

- Доля времени, в течение которого процессор активен (не находится в режиме ожидания).

1: 50%

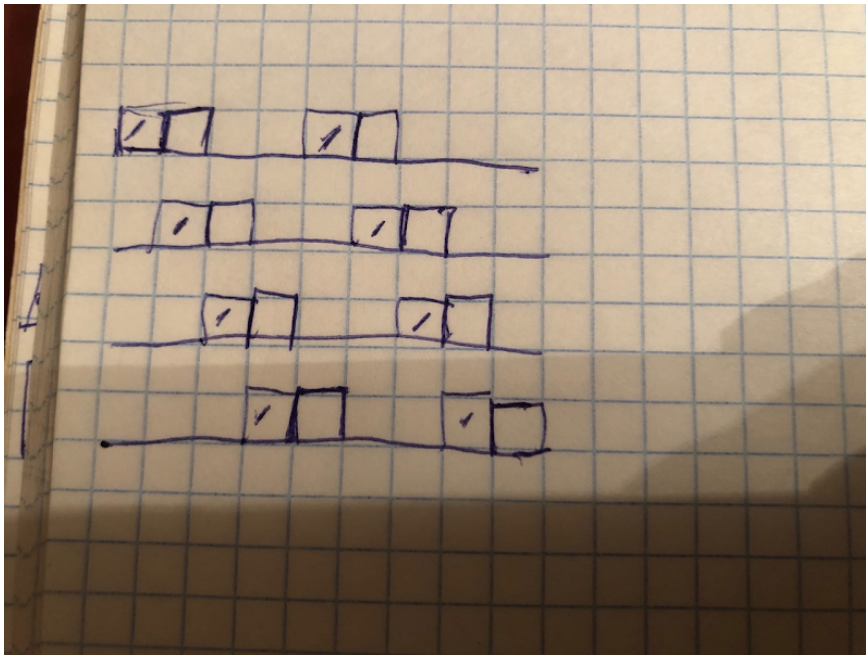
2: 100%

4: 100%

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

а. В течение первой половины периода выполняется ввод-вывод, а в течение второй - работа процессора.

б. В течение первой и четвертой четвертей выполняется ввод-вывод, а в течение второй и третьей - работа процессора.



3.2 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.

A	B	C	D	E	F	G	
	p1	p3	p5	p7	p8		
22	wb	wb	runing/runable	wb	любой кроме exit		
37	runing/runable	runing/runable	ws	wb	любой кроме exit		
47	runing/runable	runing/runable	runing/runable	wb	exit		

7.2 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?

8 бит

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

а. Сколько битов в логическом адресе?

26 бит(2^{16} страниц + 2^{10} байт в каждой странице)

б. Сколько байт в кадре?

2^{10} (равно количеству байт в странице)

в. Сколько битов физического адреса определяет кадр?

22 бита(равно количеству страниц памяти)

г. Сколько записей в таблице страниц?

2^{16} (равно количеству страниц логического адресного пространства)

д. Сколько битов в каждой записи таблицы страниц? Предполагаем, что каждая

запись таблицы страниц содержит бит корректности страницы.

23 бита(22 бита = номер кадра + 1 бит корректности)

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

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

Начальный адрес Длина (байты)

660	248
1752	422
222	198
996	604

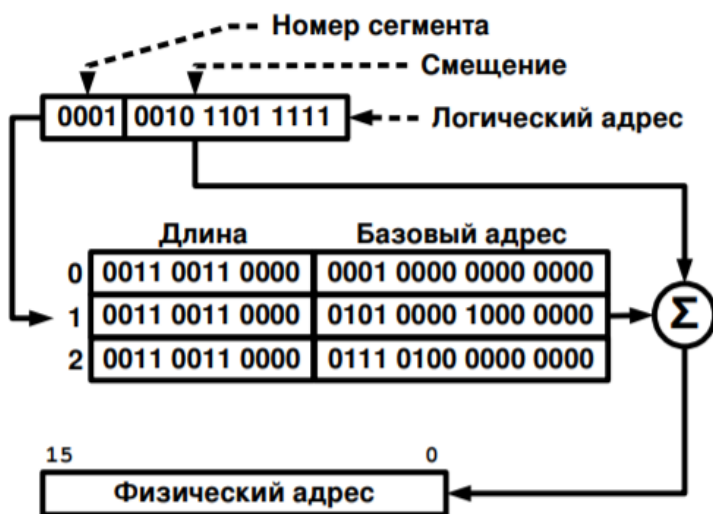
Для каждого из следующих логических адресов определите физический адрес или укажите, что заданный адрес ошибочен.

- а. 0, 198
- б. 2, 156
- в. 1, 530
- г. 3, 444
- д. 0, 222

В, смещение больше длины сегмента

формат записи:

номер записи в таблице, смещение



это считать по этой схеме, но я не могу понять как (help)

8.1 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

а. Опишите, как именно виртуальный адрес транслируется в физический адрес основной памяти.

номер кадра, соответствующий номеру страницы, умножаем на 1024(размер страницы == размер кадра) + смещение

б. Какой физический адрес (если таковой имеется) соответствует каждому из приведенных виртуальных адресов? (Вы не должны пытаться обработать прерывание из-за отсутствия страницы) .

- 1052

7196

- 2221

Pagefault

- 5499

379

б) $1052 = 1024 \cdot 1 + 28$ сопоставляется с VPN **1** в PFN 7, $(7 \cdot 1024 + 28) = \mathbf{7196}$

$2221 = 1024 \cdot 2 + 173$ сопоставляется с VPN **2** в PFN ---, **ошибка страницы**

$5499 = 1024 \cdot 5 + 379$ сопоставляется с VPN **5** в PFN 0, $(0 \cdot 1024 + 379) = \mathbf{379}$

8.4 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.

а. FIFO ("первым вошел первым вышел").

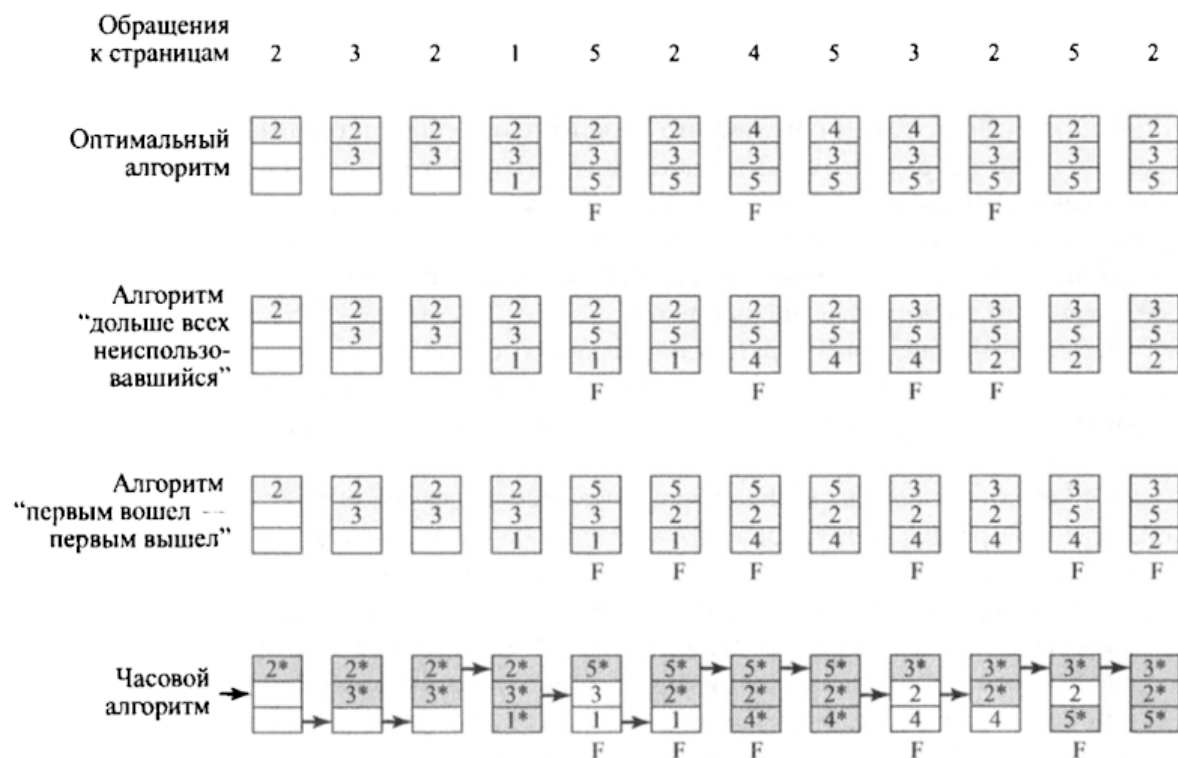
б. LRU (последний использовавшийся).

в. Часовой.

г. Оптимальный (в предположении, что последовательность обращений продолжается как 1, 2,

0, 1, 7, 0, 1).

д. Перечислите общее количество ошибок страниц и частоту промахов для каждой стратегии. Подсчитайте количество ошибок страницы, происшедших после того, как все кадры были инициализированы.



F --- прерывания обращения к странице после первоначального заполнения кадров

Рис. 8.14. Поведение четырех алгоритмов замещения страниц

(там ошибка с цифрами, отличаются от задания, но суть та же)

8.6 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.

а. 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).

б. Выполните то же задание для алгоритма "первым вошел - первым вышел".

в. Сравните результативности обращения к основной памяти, вычисленные в первых двух заданиях, и прокомментируйте эффективность использования указанных алгоритмов применительно к данной последовательности обращений.

см предыдущее

8.10 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?

1 уровень 1024 записи на 4кб => уже можем разметить ($2^{10} * 2^{12}$)

2 уровень 1024 (ссылается на уровень выше) * 1024 записи * 4кб = ($2^{10} * 2^{10} * 2^{12}$)

....

6 уровень $(2^{10})^6 * 4кб = 2^{12} * (2^{10})^6 = 2^{72}$ адресов

Ответ: 6

8.17 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.

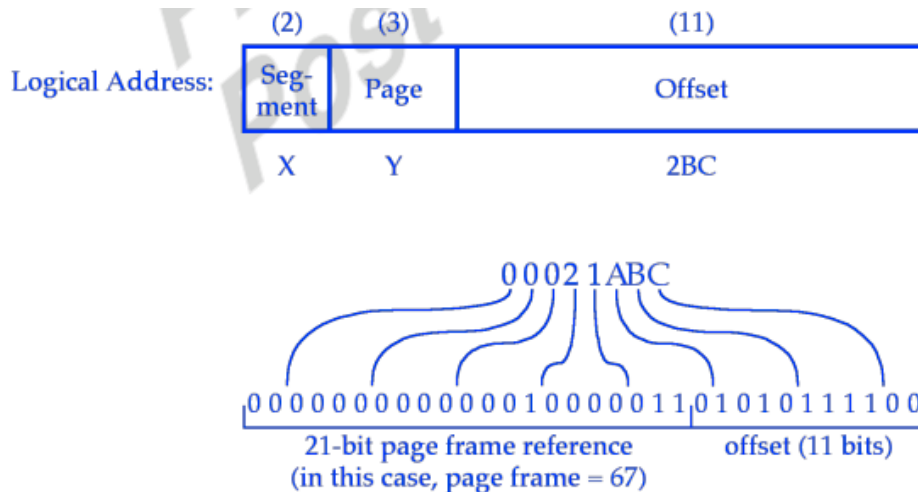
а. Чему равен максимальный объем каждого сегмента?

16 Кбайт

б. Каково максимальное логическое адресное пространство одного задания?

2^{16} байт (16кбайт * 4 сегмента = 64кб)

в. Предположим, что рассматриваемое задание обратилось к ячейке памяти с физическим адресом 00021ABC. Каков формат генерируемого для этого логического адреса? Каково максимально возможное физическое адресное пространство в этой системе?



9.16. Пять пакетных заданий, от А до Е, 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.

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

ABCDE

ABCDE

ABCDE

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

ABDE

ABDE

ABDE

ABE

ABE

ABE

AE

AE

AE

A

A

A

A = 45 мин

B = 35 мин

C = 13 мин

D = 26 мин

E = 42 мин

б. Планирование с учетом приоритетов.

BEACD

B = 9 мин

E = 21 мин

A = 36 мин

C = 39 мин

D = 45 мин

в. FCFS при запуске процессов в следующем порядке: 15, 9, 3, 6 и 12.

ABCDE

A = 15 мин

B = 24 мин

C = 27 мин

D = 33 мин

E = 45 мин

г. Первым выполняется самое короткое задание.

CDBEA

C = 3 мин

D = 9 мин

B = 18 мин

E = 30 мин

A = 45 мин

11.7 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

Запись идет на дорожки, которая бьется по секторам

$300\,000 / 4(512 / 120) = 75000$ секторов

782 дорожек

8 поверхностей

1 диск

11.12 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?

0 - 800gb

1 - 400gb

3 - 600gb

4 - 600gb

5 - 600gb

6 - 400gb

Таблица 11.4. Уровни RAID

Категория	Уровень	Описание	Требуемое количество дисков ¹	Доступность данных	Пропускная способность передачи больших данных	Скорость запросов малого ввода-вывода
Расщепление	0	Без избыточности	N	Меньше, чем у одного диска	Очень высокая	Очень высокая для чтения и записи
Отражение ²	1	Отражение	$2N$	Больше, чем у RAID 2, 3, 4 и 5, но меньше, чем у RAID 6	Для чтения больше, чем у одного диска; для записи сравнима с одним диском	Для чтения почти вдвое больше, чем у одного диска; для записи сравнима с одним диском
Параллельный доступ	2	Избыточность с кодами Хэминга	$N+m$	Гораздо выше, чем у одного диска, сравнимо с RAID 3, 4 и 5	Наибольшая среди всех перечисленных альтернатив	Почти вдвое больше, чем у одного диска
	3	Четность с чередующимися битами	$N+1$	Гораздо выше, чем у одного диска, сравнимо с RAID 2, 4 и 5	Наибольшая среди всех перечисленных альтернатив	Почти вдвое больше, чем у одного диска
Независимый доступ	4	Четность с чередующимися блоками	$N+1$	Гораздо выше, чем у одного диска, сравнимо с RAID 2, 3 и 5	Для чтения аналогична RAID 0; для записи значительно меньше, чем у одного диска	Для чтения аналогична RAID 0; для записи значительно меньше, чем у одного диска
	5	Распределенная четность с чередующимися блоками	$N+1$	Гораздо выше, чем у одного диска, сравнимо с RAID 2, 3 и 4	Для чтения аналогична RAID 0; для записи меньше, чем у одного диска	Для чтения аналогична RAID 0; для записи в общем случае меньше, чем у одного диска
	6	Двойная распределенная четность с чередующимися блоками	$N+2$	Наибольшая среди всех перечисленных альтернатив	Для чтения аналогична RAID 0; для записи меньше, чем у RAID 5	Для чтения аналогична RAID 0; для записи значительно меньше, чем у RAID 5

¹ N — количество дисков данных; m пропорционально $\log N$.

² Иногда в литературе можно встретить перевод "mirroring" как "зеркалирование". — *Примеч. пер.*

12.7 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.

Размер файла / суммарный размер минимально необходимого количества блоков

$$1 - (41600 / (3 \cdot 16 \cdot 1024)) = \mathbf{0.1536}$$

$$1 - (640000 / (40 \cdot 16 \cdot 1024)) = \mathbf{0.0234375}$$

$$1 - (4064000 / (249 * 16 * 1024)) = 0.003827811244979884$$

12.12. 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?

прямой уровень = 12 по 1кб

1 уровень = 256 * 1 кб

2 уровень = 256 * 256 * 1 кб

3 уровень = 256 * 256 * 256 * 1 кб

16843020 кб

12.13. 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).

а. Какой максимальный размер файла, поддерживаемый в этой системе?

2048 адресов в блоке ($2^{13} / 4 = 2^{11}$)

прямой = 12 * 8 кбайт

1 уровень $2048 * 8$ кбайт

2 уровень $2048 * 2048 * 8$ кбайт

3 уровень $2048 * 2048 * 2048 * 8$ кбайт

68753047648 кбайт

б. Какой максимальный размер раздела файловой системы, поддерживаемого в этой системе?

$2^{24} * 2^{13}$ байт = **128 гб** диск

в. 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?

2

(так как первый уровень косвенности, сначала заходим в нужный блок, откуда получаем ссылку на нужный нам блок, дальше читаем уже нужный блок)

Дополнительно:-----

1.13 В компьютере есть кеш, основная память и диск, выступающий в роли виртуальной

памяти. Если запрашиваемое слово находится не в кеше, а в основной памяти, для его

загрузки в кеш 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.

Среднее время доступа:

зультате получим

$$T_s = H \times T_1 + (1 - H) \times (T_1 + T_2) = T_1 + (1 - H) \times T_2, \quad (1.1)$$

где

T_s — среднее (системное) время доступа,
 T_1 — время доступа к М1 (например, кешу, дисковому кешу),
 T_2 — время доступа к М2 (например, основной памяти, диску),
 H — результативность поиска (доля ссылок на данные в М1).

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	$(0.1)(0.4) = 0.04$	$12\text{ms} + 60 + 20 = 12,000,080$

So the average access time would be:

$$\text{Avg} = (0.9)(20) + (0.06)(80) + (0.04)(12000080) = 480026 \text{ ns}$$

-9-

Вариант 23

Рассмотрим оперативную память (м) и КЭШ (с), характеризующуюся следующими параметрами:

$T_c = 100 \text{ нс};$

$T_m = 1200 \text{ нс};$

$C_c = 0,01 \text{ у.е./бит};$

$C_m = 0,001 \text{ у.е./бит}.$

1. Сколько стоит 1 Мбайт оперативной памяти?
2. Сколько стоит 1 Мбайт оперативной памяти, выполненной по технологии КЭШа?
3. Какова результативность поиска h , если эффективное время доступа на 10 % больше, чем время доступа к КЭШу?

x - результативность поиска

$$1,1 T_c = x T_c + (1-x)(T_c + T_m)$$

$$110 = x \cdot 100 + (1-x) \cdot (1300)$$

$$110 = 100x - 1300x + 1300$$

$$1200x = 1190$$

$$x = 1190/1200 = \mathbf{119/120}$$

=====

Рассмотрим страничное логическое адресное пространство, состоящее из 22-х страниц по 2 Кбайта каждая, отображенное на 2 Мбайтовом физическом пространстве.

Чему равна длина и ширина таблицы страниц (без учета битов прав доступа).

2Мбайтовое пространство = 2^{21} адресов \Rightarrow 21 битный физический адрес, будет указыв

Кол-во страниц: $2^{21}/2^{11} = 2^{10} \Rightarrow$ 10 бит на номер страницы

а когда больше 2 процессов, максимально за цикл на процессор попадает 2 таска

Это в первую очередь значение того, какую часть из всех задач мы успеваем выполнить за T . Кстати даже когда какая то из задач ничего не делает в какой то из циклов и ждёт своей очереди, тоже считается работой, поскольку входит в время выполнения отдельно взятой задачи из формул выше. Грубо говоря ответом здесь будет "количество задач, выполняющихся одновременно/среднее время выполнения одного задания"