

1. В ОЗУ 16-разрядного компьютера используется контроль целостности данных по четности. Описать возможную структуру ячейки памяти и ее побитовое содержимое для случая хранения в машинном слове восьмеричного числа 17735_8 .

Ответ: в 16-разрядном компьютере ячейка памяти содержит 16 бит данных, бит паритета (тег) – дополнительный, он в данные не входит.

Структура ячейки памяти: 001 111 111 011 101 – это данные, и еще один бит паритета (приписывается справа или слева) равен 1 – он вычисляется как сумма по модулю 2 (хор) всех битов данных.

2. Пусть дано восьмеричное число 173357_8 , являющееся адресом оперативной памяти, расслоенной по 16 банкам. Банку с каким номером принадлежит заданный адрес?

Ответ: 13. В схеме расслоения памяти последовательные адреса размещаются в последовательных банках памяти. При 16 банках памяти за номер банка будут отвечать младшие 4 бита адреса. Они равны 1111 (от восьмеричной пятёрки в двоичном виде 101 берется последняя единица, из восьмеричной семерки получается двоичное 111), то есть 15 в десятичной системе. Банки нумеруются всегда с 0.

3. Дан 32-разрядный IP-адрес, имеющий в восьмеричном представлении вид 23171171543_8 . Определить к какому классу относится данный IP адрес и номер сети (в восьмеричном представлении), к которой относится IP адрес.

Ответ: $23171171543_8 = 010\ 011\ 001\ 111\ 001\ 001\ 111\ 001\ 101\ 100\ 011_2$

Старшие биты адреса: 10, это сеть класса B, номер сети – это 14 следующих бит, номер хоста – оставшиеся 16 бит. Тогда номер сети $011\ 001\ 111\ 001\ 00_2 = 14744_8$.

4. Пусть некоторый процесс с pid=A породил два сыновних процесса с pid=B и pid=C, реализованный программой на Си:

```
int main (int argc, char **argv) /* pid=A */
{
    if (fork()==0){ /* pid=B */
        printf ("%d %d\n", getppid(),getpid());
        exit(0);
    }
    if (fork()==0){ /* pid=C */
        printf ("%d\n",getpid());
        exit(0);
    }
    return 0;
}
```

Считаем, что printf работает атомарно и обращения ко всем системным вызовам срабатывают успешно. Перечислить все возможные комбинации значений, которые попадут на стандартное устройство вывода в результате выполнения данной программы.

Ответ: Процесс А не ждет процесс В (отсутствует wait()), поэтому А может завершиться как до завершения В (тогда осиротевший В станет потомком процесса init с pid=1), так и после. Никакой синхронизации между братьями В и С нет. Поэтому возможны четыре варианта ответа:

1	2	3	4
А В	1 В	С	С
С	С	А В	1 В

5. Пусть дана файловая система Unix System V и в ней утеряна информация суперблока. Предложить последовательность действий, позволяющую восстановить содержимое файлов данной файловой системы. Считаем, что до потери суперблока содержимое файловой системы было корректным. Размер суперблока, размер и структура индексного дескриптора известны.

Ответ: структура файловой системы версии System V:

Суперблок	Область индексных дескрипторов	Блоки файлов
-----------	--------------------------------	--------------

Размер области индексных дескрипторов хранится в суперблоке. При потере информации суперблока данное значение теряется. Для восстановления содержимого файлов необходимо определить границу между областью индексных дескрипторов и областью блоков файлов. От начала области индексных дескрипторов итеративно повторяем следующие действия. Считываем очередной ID. Проверяем содержимое поля «ссылки на данный ID каталогов файловой системы». Если это поле равно нулю (это означает, что ID свободен), переходим к следующему ID. В противном случае последовательно просматриваем 13 элементов, описывающих адресацию блоков файла (до завершения): номера блоков с прямой адресацией (10 штук), номера блоков с косвенной адресацией 1,2 и 3-х уровневой. В случае, если получен некорректный номер блока, завершаем просмотр (область индексных дескрипторов закончилась).

Альтернативой проверки поля со ссылками может являться проверка содержимого поля «тип файла»: если оно является некорректным, то это так же означает, что область индексных дескрипторов закончилась.

6. Описать программу на языке Си, которая реализует процесс А, принимающий из командной строки полное имя существующего непустого бинарного файла (условное имя NameF), содержащего значения типа int. Процесс А связывается со своим сыновним процессом посредством неименованного канала и последовательно передает ему все целочисленные значения, хранящиеся в файле NameF (в сыновнем процессе данный файл не должен быть открыт). Процесс-сын находит номер максимального полученного значения (нумерация с 0), и выводит его на стандартное устройство вывода. Процесс-отец завершается после завершения процесса-сына. Считаем, что все обращения ко всем системным вызовам срабатывают успешно.

Ответ: По условию задачи печатать pid не обязательно, главное – правильно обменяться. Обмен происходит по двум каналам одновременно, так что действительно порядок обмена получается произвольным.

```
#include <stdio.h>
#include <unistd.h>
int main ()
```

```

{
    int fd[2];
    int tmp=0, max=0, pos=-1, count=0,f;
    pipe(fd);
    if (fork()==0){
        close(fd[1]);
        while(read(fd[0],&tmp, sizeof(tmp)>0){
            if((pos==-1)|| (tmp>max)){
                max=tmp;
                pos=count;
            }
            count++;
        }
        close(fd[0]);
        printf("%d\n",pos);
        exit(0);
    }

    close(fd[0]);
    f=open(argv[1],O_RDONLY);
    while(read(f,&tmp, sizeof(tmp)>0){
        write(fd[1],&tmp, sizeof(tmp));
    }
    close(fd[1]);
    wait(NULL);
    return 0;
}

```