

№1	№2	№3	№4	Σ	ФИО экзаменатора (письмен.)

ФИО _____ Группа _____ ВАРИАНТ № 3 11.12.2021

В ответах указывается № вопроса и дается информация, относящаяся исключительно к заданному вопросу. Писать аккуратно, разборчиво.

1	В компьютере используется 8-сегментная модель организации памяти. Написать функцию <code>unsigned int VirtIntoPhys (segment *SegTable, unsigned int VirtAdr)</code> , реализующую преобразование виртуального сегментного адреса (передается в виде параметра <code>VirtAdr</code> в функцию) некоторого процесса в физический (возвращается как результат обращения к функции). Считаем, что все сегменты процесса размещены в физической памяти. Ситуацию обнаружения ошибки при преобразовании виртуального адреса реализовать завершением работы процесса с программным кодом завершения, равным 25. Считаем, что таблица сегментов реализуется в виде массива структур и передается в программу по указателю <code>*SegTable</code> . Описать и прокомментировать в программе все необходимые структуры данных, при учете того, что в машине используются виртуальные и физические адреса, равные по размеру <code>int</code> (32 бита). (Вес задачи: 2 балла)
2	Написать программу, моделирующую работу Raid 0, построенного из 4-х дисков (вместо реальных дисков используются файлы с именами «Disk0», «Disk1», ... «Disk3»). Размер блока – <code>BlockSize</code> (определенная в программе константа, измеряется в байтах). Нумерация блоков – от нуля. В программе должны быть: функция <code>void Ini_Raid0(void)</code> – инициализация (открытие файлов «Disk0», «Disk1», ... «Disk3» – считаем, что они существуют, доступны и нужного размера), а также функция <code>void Read_Raid0(int num, char *buf)</code> – запрос на чтение блока с номером <code>num</code> в буфер по указателю <code>*buf</code> . Функцию записи и функцию <code>main</code> реализовывать не нужно. (Вес задачи: 3 балла)
3	Что будет выведено на экран? Прокомментировать, почему? Если возможны несколько вариантов – привести все. Предполагается, что обращение к функции вывода на экран прорабатывает атомарно и без буферизации. Все системные вызовы прорабатывают успешно. Подключение заголовочных файлов опущено. <pre>int main() { int pid; int fd[2]; char c = 'a'; pipe(fd); if((pid = fork()) > 0) { read(fd[0], &c, 1); kill(pid, SIGKILL); wait(NULL); } else { putchar(c); c = 'b'; write(fd[1], &c, 1); c = 'c'; } putchar(c); return 0; }</pre> (Вес задачи: 2 балла)
4	Сколько раз система обратится к содержимому индексных дескрипторов при вызове: <code>open("/dir1/dir2/dir3/file", O_RDONLY)</code> ? Прокомментировать, почему? Считаем, что ни один из элементов пути к файлу не является символической ссылкой. (Вес задачи: 1 балл)