

Лекции 2-3

Program №1

Пример программы динамического выделения памяти для массива.

Программа запрашивает длину массива. Затем выделяется память с помощью функции malloc. При выделении памяти возвращается указатель, который будет равен NULL, если память не выделилась. Если массив инициализирован, то выделенная память освобождается.

Тестирование 1

```
alex@192:~/Рабочий стол/Lec2,3> ./Program_1
Enter length of array: -10
Error: can't allocate memoryalex@192:~/Рабочий стол/Lec2,3> █
```

Тестирование 2

```
alex@192:~/Рабочий стол/Lec2,3> ./Program_1
Enter length of array: 1
All fine
alex@192:~/Рабочий стол/Lec2,3> █
```

Program №2

Пример программы для чтения данных из файла.

В начале выделяем память для массива char с помощью функции calloc. Затем создаем дескриптор и открываем файл для чтения. Если файла не будет, то он будет создан. Выводим значение дескриптора файла.

После читаем из файла 10 байт и записываем в переменную sz количество байт, которое удалось прочитать. Далее записываем в конец массива, символ конца строки. В конце программы закрываем файл.

Тестирование

```
alex@192:~/Рабочий стол/Lec2,3> ./Program_2
fd = 3/ncalled read( 3, c, 10). returned that 0 bytes were read.
closed the fd.
alex@192:~/Рабочий стол/Lec2,3> █
```

Program №3

Пример программы системного вызова fork().

В начале вызывается системный вызов `fork()`. После этого момента программа делится на двое и в результате вызова функции появляется один ребенок. С помощью оператора `switch-case` мы определяем, каким является процесс. Если ошибка (`case -1:`), если ребенок (`case 0:`) или если родитель (`default:`). Как правило, процесс потомка всегда больше, чем у родителя, а определяется это с помощью функции `getpid()`.

Тестирование

```
alex@192:~/Рабочий стол/Lec2,3> ./Program_3
my pid = 91879, returned pid = 91880
my pid = 91880, returned pid = 0
alex@192:~/Рабочий стол/Lec2,3> █
```