

## **Практическое задание № 2. Консоль управления моделью Simple Computer. Текстовая часть.**

### *Цель работы*

Изучить принципы работы терминалов ЭВМ в текстовом режиме. Понять, каким образом кодируется текстовая информация и как с помощью неё можно управлять работой терминалов. Разработать библиотеку функций `myTerm`, включающую базовые функции по управлению текстовым терминалом (очистка экрана, позиционирование курсора, управления цветом). Начать разрабатывать консоль управления Simple Computer (вывести на экран текстовую часть).

### *Задание на лабораторную работу*

1. Прочитайте главу 5 практикума по курсу «Организация ЭВМ и систем». Обратите особое внимание на параграфы 5.4 и 5.5. Изучите страницу `man` для команды `infocmp`, базы `terminfo`, функции `ioctl`.
2. Откройте текстовый терминал и запустите оболочку `bash` (оболочка запускается автоматически). Используя команду `infocmp`, определите (и перепишите их себе) `escape`-последовательности для терминала, выполняющие следующие действия:
  - очистка экрана и перемещение курсора в левый верхний угол (`clear_screen`);
  - перемещение курсора в заданную позицию экрана (`cursor_address`);
  - задание цвета последующих выводимых символов (`set_a_background`);
  - определение цвета фона для последующих выводимых символов (`set_a_foreground`);
  - скрывание и восстановление курсора (`cursor_invisible`, `cursor_visible`).
3. Используя оболочку `bash`, команду `echo -e` и скрипт<sup>1</sup>, проверьте работу полученных последовательностей. Символ `escape` задается как `\033` или `\E`. Например – `echo -e "\033[m`". Для проверки сформируйте последовательность `escape`-команд, выполняющую следующие действия:
  - очищает экран;
  - выводит в пятой строке, начиная с 10 символа Ваше имя красными буквами на черном фоне;
  - в шестой строке, начиная с 8 символа Вашу группу зеленым цветом на белом фоне;
  - перемещает курсор в 10 строку, 1 символ и возвращает настройки цвета в значения «по умолчанию».
4. Разработать следующие функции:
  - `int mt_clrscr (void)` - производит очистку и перемещение курсора в левый верхний угол экрана;
  - `int mt_gotoXY (int, int)` - перемещает курсор в указанную позицию. Первый параметр номер строки, второй - номер столбца;
  - `int mt_getscreensize (int * rows, int * cols)` - определяет размер экрана терминала (количество строк и столбцов);
  - `int mt_setfgcolor (enum colors)` - устанавливает цвет последующих выводимых символов. В качестве параметра передаётся константа из созданного Вами перечислимого типа `colors`, описывающего цвета терминала;
  - `int mt_setbgcolor (enum colors)` - устанавливает цвет фона последующих выводимых символов. В качестве параметра передаётся

---

<sup>1</sup> Скрипт – это текстовый файл, содержащий команды оболочки. Запускается на выполнение командой `bash имя_файла`.

константа из созданного Вами перечислимого типа `colors`, описывающего цвета терминала.

Все функции возвращают 0 в случае успешного выполнения и -1 в случае ошибки. В качестве терминала используется стандартный поток вывода.

5. Оформите разработанные функции как статическую библиотеку `myTerm`. Подготовьте заголовочный файл для неё.

#### *Защита лабораторной работы*

Для защиты лабораторной работы необходимо подготовить программу, демонстрирующую использование созданной библиотеки функций (сборка программы с библиотекой, использование заголовочного файла, примеры вызовов каждой функции, проверка корректности работы функций при различных входных значениях), а также программу, выводящую на экран согласно рисунку 1 содержимое оперативной памяти, регистров и назначение клавиш.

#### *Контрольные вопросы*

1. Взаимодействие с устройствами в Linux. Специальные файлы устройств.
2. Функции `open`, `close`, `read`, `write`.
3. Терминалы. Типы терминалов. Эмуляция терминала. Режимы работы.
4. Управление терминалом. Команды. Низкоуровневое управление.
5. Что такое `escape`-последовательность?
6. Как определить `escape`-последовательности для терминала?