Федеральное агентство связи Федеральное государственное бюджетное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

Лабораторная работа 5

“Консоль управления моделью Simple Computer. Клавиатура. Обработка нажатия клавиш. Неканонический режим работы терминала”

Выполнили:

Студенты группы ИП-813

Бурдуковский И.А

Стояк Ю.К

Проверил: Преподаватель кафедры ВС Токмашева Елизавета Ивановна

Новосибирск, 2020

## Цель работы

Изучить устройство клавиатуры и принципы обработки нажатия клавиш в текстовом терминале. Создать «распознаватель» нажатой клавиши по формируемой последовательности символов. Разработать библиотеку myReadkey. Доработать интерфейс консоли управления Simple Computer так, чтобы можно было изменять значения ячеек памяти и регистров

## Выполнение работы

За «распознаватель» нажатой клавиши по формируемой последовательности символов отвечает функция **int rk\_readKey(enum keys \*key)** библиотеки **myReadkey**, где параметр key переменная, запоминающая нажатую клавишу. Она работает следующим образом. Функция переводит терминал в неканонический режим при помощи функции **rk\_mytermregime** библиотеки **myReadkey**. После чего функция **read** ожидает нажатие клавиши от пользователя и записывает код нажатой клавиши в массив **char buf**. Если была нажата специальная клавиша, то в массив **buf** будет записан код специальной клавиши (например \n \033). В других случаях **buf** будет хранить один символ обычной клавиши (если была нажата клавиша ‘h’, то в **buf** будет записано ‘h’). После при помощи условных операторов **if** и **switch** определяется нажатая клавиша и присваивается переменной **key**.

int rk\_readkey(enum keys \*key)

{

termios orig\_conf;

char buff[16] = "\0";

tcgetattr(1,&orig\_conf);

rk\_mytermregime(0, 0, 1, 0, 1);

read(1,buff,15);

tcsetattr(1,TCSANOW,&orig\_conf);

if (!strncmp(buff, "q",1))

\*key = KEY\_q;

if (!strncmp(buff,"l",1))

\*key=KEY\_l;

if (!strncmp(buff,"s",1))

\*key=KEY\_s;

if (!strncmp(buff,"r",1))

\*key=KEY\_r;

if (!strncmp(buff,"t",1))

\*key=KEY\_t;

if (!strncmp(buff,"i",1))

\*key=KEY\_i;

if (!strncmp(buff,"\E[15~",5))

\*key=KEY\_f5;

if (!strncmp(buff,"\E[17~",5))

\*key=KEY\_f6;

if (!strncmp(buff,"\E[A",3))

\*key=KEY\_up;

if (!strncmp(buff,"\E[B",3))

\*key=KEY\_down;

if (!strncmp(buff,"\E[D",3))

\*key=KEY\_left;

if (!strncmp(buff,"\E[C",3))

\*key=KEY\_right;

if (!strncmp(buff,"\n",1))

\*key=KEY\_enter;

return 0;

}

Функция **int rс\_mytermsave ()** - сохраняет текущие параметры терминала. При помощи функции **tcgetattr(1, &termsave)** получаем параметры, связанные с объектом, на который ссылается дескриптор, и сохраняем их в структуре **termsave**.

int rc\_mytermsave(){

tcgetattr(1, &termsave);

return 0;

}

Функция **int rk\_mytermrestore ()** – загружает сохраненные параметры из структуры termsave.

int rk\_mytermrestore(){

tcsetattr(1,TCSAFLUSH,&termsave);

return 0;

}

При помощи **tcsetattr** **(1, TCSAFLUSH, &termsave)** изменяются параметры, связанные с терминалом. Параметры, которые необходимо установить хранятся в структуре **termsave.**

Функция **int rk\_mytermregime (int regime, int vtime, int vmin, int echo, int sigint)** - переключает терминала между режимами.

Параметры:

* + **ICANON** – при значении 1, устанавливает неканонический режим работы терминала.
  + **VTIME** – время ожидания появления хотя бы одного символа в очереди.
  + **VMIN** – количество символов, которые должны быть в очереди, чтобы вызов **read** завершился.
  + **ECHO** - включает локальное отображение вводимых символов
  + **ISIG** – генерация сигналов, когда принимаются любые символы из INTR, QUIT, SUSP, DSUSP.

Структура функции следующая:

Сохраняются текущие параметры терминала и создаётся новая структура **termsave**, в которую записываются новые параметры принятые функцией **rk\_mytermregime.** А именно устанавливаются константы флагов **c\_lflag**: ICANON, ISIG, ECHO. Устанавливается время ожидания и количество символов в очереди в **c\_cc** массив. При помощи функции **tcsetattr** в терминал загружаются новые параметры.

int rk\_mytermregime(int regime, int vtime, int vmin, int echo, int sigint)

{

if (tcgetattr(1,&mode)!=0)

return -1;

if (regime==1) {

mode.c\_lflag|=ICANON;

mode.c\_cc[VTIME]=0;

mode.c\_cc[VMIN]=1;

mode.c\_lflag|=ECHO;

mode.c\_lflag|=ISIG;

return -1;

}

if (regime==0)

{

mode.c\_lflag&=~ICANON;

mode.c\_cc[VTIME]=vtime;

mode.c\_cc[VMIN]=vmin;

if (echo==1)

mode.c\_lflag|=ECHO;

else if (echo==0)

mode.c\_lflag&=~ECHO;

else

return -1;

if (sigint==1)

mode.c\_lflag|=ISIG;

else if (sigint==0)

mode.c\_lflag&=~ISIG;

else

return -1;

}

if (tcsetattr(1, TCSANOW, &mode)!=0)

return -1;

return 0;

}

## Демонстрация работы доработанного интерфейса консоли управления Simple Computer



