|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБАРАТОРНОЙ РАБОТЫ №3**

по дисциплине: «Операционные системы»

|  |  |  |
| --- | --- | --- |
| Студент |  | Беркович Евгений Сергеевич |
| Группа |  | РК6-55Б |
| Тип задания |  | Тренажёр клавиатуры |
| Вариант |  | 02 |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_ **Беркович Е.С.**\_ \_

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_ **Грошев С. В.** \_\_

*подпись, дата фамилия, и.о.*

*Москва, 2023 г.*

**Техническое задание**

Требуется, разработать программу, которая реализует терминальный интерфейс тренажёра клавиатуры для консольного ввода текстовые строк по образцу. Образцы строк должны выбираться случайным образом из текстового  
файла, имя которого передается программе через аргумент командной строки. Таким образом, после ввода очередной строки появляется следующая строка как образец для ввода.

При этом должны исключаться пустые строки, символы табуляции и   
пробелы.

**Приложение №1**

**Содержимое файла terminal.c**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <termios.h>  #include <time.h>  #include <unistd.h>  int textmode(int);  int randline(FILE\* file);  int getch();  static char pattern[80];  static char buffer[80];  static int LengthUsed;  int textmode(int mode) {  static struct termios con[2];  if (mode > 0) return (tcsetattr(0, TCSAFLUSH, &con[1]));  tcgetattr(0, &con[0]);  tcgetattr(0, &con[1]);  con[0].c\_lflag &= ~(ICANON | ECHO | ISIG);  con[0].c\_iflag &= ~(ISTRIP | IXOFF | IXANY | IXON);  con[0].c\_oflag |= CS8;  con[0].c\_cc[VMIN] = 2;  con[0].c\_cc[VTIME] = 1;  return (tcsetattr(0, TCSAFLUSH, &con[0]));  }  int getch() {  unsigned char c[2];  static int len = 0;  /\*if(len > 1) {  c[0] = len;  len = 0;  return(c[0]);  }  \*/  c[0] = c[1] = 0;  if ((len = read(0, c, 2)) < 2) return (c[0]);  if (c[0] == 27) c[0] = 0;  /\* len = c[1]; \*/  ungetc(c[1], stdin);  return (c[0]);  }  int randline(FILE\* file) {  // memset(pattern,NULL,80);  int l[2];  l[0] = 1;  l[1] = 1;  int i = 0;  char c;  int strammount = 0;  char line[256]; /\* or other suitable maximum line size \*/  while (fgets(line, sizeof(line), file) != NULL) /\* read a line \*/  {  strammount++;  }  int strnumber = rand() % strammount + 1;  write(1, "str N", 6); // Make space for sprintf to work in.  LengthUsed = sprintf(buffer, "%d", strnumber); // Format n in decimal.  write(1, buffer, LengthUsed);  write(1, ":", 1);  rewind(file);  strammount = 0;  while ((strammount + 1) != strnumber) {  fgets(line, sizeof(line), file);  strammount++;  }  while (l[1]) {  c = getc(file);  switch (c) {  case ' ':  break;  case '\n':  l[1] = 0;  break;  case EOF:  l[1] = 0;  break;  default:  if (i < 80) {  pattern[i] = c;  i++;  } else {  l[1] = 0;  }  break;  }  }  return i;  }  int main(int argc, char\* argv[]) {  srand(time(NULL));  FILE\* text;  int len;  int i = 0;  int err = 0;  unsigned char c;  len = 6; /\*atoi(argv[1]);\*/  if (!argv[1]) {  write(1, "NO SOURCE FILE ENTERED\n", 23);  } else if ((text = fopen(argv[1], "r")) == NULL) {  write(1, "SOURCE FILE WAS REJECTED\n", 26);  } else {  textmode(0);  int blank = 1;  int bl=1;  while (blank) {  len = randline(text);  write(1, pattern, len);  write(1, "\n", 1);  while (i < len && bl) {  switch (c = getch()) {  case 0:  c = '\007';  switch (getch()) {  case 67:  c = pattern[i];  write(1, "per", 3);  break;  case 68:  if (i == 0) break;  i--;  write(1, "\b", 1);  continue;  default:  break;  }  break;  case 27:  bl=0;  blank = 0;  break;  default:  if (c != pattern[i]) c = '\007';  break;  }  if(c!=27)  (c == '\007') ? err++ : i++;  write(1, &c, 1);  }  if (len == 0) {  write(1, "ignored", 8);  } else if (i!=0){  write(1, "\n", 1);  write(1,"errors:",8);  LengthUsed = sprintf(buffer, "%d", err); // Format n in decimal.  write(1, buffer, LengthUsed);  write(1," ",1);  write(1,"total:",7);  i+=err;  LengthUsed = sprintf(buffer, "%d", i); // Format n in decimal.  write(1, buffer, LengthUsed);  }  write(1, "\n", 1);  i=0;  err=0;  }  }  textmode(1);  return (err);  } |

**Приложение №2**

**Результат работы программы:**

Программа принимает аргумент командной строки: имя файла со строками и случайным образом выбирает строку-образец;

Рассмотрены краевые ситуации: название не было передано в качестве аргумента; файл, переданный в качестве аргумента отсутствует/испорчен/не имеет достаточных прав для использования в программе.

