



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпровський національний університет
залізничного транспорту ім. академіка В. Лазаряна

Кафедра КІТ

ЛАБОРАТОРНА РОБОТА № 10
з дисципліни: «Операційні системи»
на тему: «Налаштування параметрів термінального інтерфейсу»

Виконав: студент групи ПЗ1712

Нікольський О.В.

Приняла: Нежуміра О.І.

Дніпро, 2020

Тема. Налаштування параметрів термінального інтерфейсу.

Завдання. Розробити програму зчитування та налаштування параметрів термінального інтерфейсу (перевірити та порівняти роботу терміналу в канонічному та неканонічному режимі. Змінити спосіб реагування на термінальні сигнали. Змінити значення символів керування).

Символи керування: EOF, KILL, QUIT.

Значення параметрів в неканонічному режимі: MIN=10, TIME=0.

Опис специфікацій

prog.cpp

У розробленій програмі користувач обирає, в якому режимі буде працювати поточна сесія терміналу. Реалізована заміна клавішних сполучень для керуючих символів EOF, KILL, QUIT (для канонічного режиму); Замінені значення параметрів в неканонічному режимі — тепер запис відбудеться лише після того, як у черзі введення буде не менше 10 байт.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <termios.h>
#include <signal.h>
#include <cstring>
#include <iostream>
/*
    Заменить символы управления - EOF, KILL, QUIT
    Значения параметров в неканоническом режиме - MIN=10, TIME=0
*/
struct termios saved_attributes;
struct termios term;
void ResetInputMode(){
    tcsetattr(0, TCSANOW, &saved_attributes);
}
bool Canonical(){
    std::cout << "Canonical mode" << std::endl;
    struct termios term;
    if(!isatty(0)){
        perror("isatty()");
        return false;
    }
    tcgetattr(0, &saved_attributes);
    memcpy(&term, &saved_attributes, sizeof(saved_attributes));
    term.c_cc[VEOF] = '\001'; // Символ EOF, Ctrl-D заменяем на Ctrl-A
    term.c_cc[VKILL] = '\002'; // Вся текущая вводимая строка уничтожается, Ctrl-U заменяем на Ctrl-B
    term.c_cc[VINTR] = 0; // Отключаем прерывание программы по Ctrl-C
    term.c_cc[VQUIT] = '\003'; // Ввод символа QUIT вызывает посылку сигнала SIGQUIT всем
    процессам основной группы процессов, Ctrl-\ заменяем на Ctrl-C
    tcsetattr(0, TCSAFLUSH, &term);
```

```

    return true;
}
bool NonCanonical(){
    std::cout << "Non-canonical mode" << std::endl;
    if(!isatty(0)){
        perror("isatty()");
        return false;
    }
    tcgetattr(0, &saved_attributes);
    term = saved_attributes;
    term.c_lflag &= ~(ICANON);
    term.c_lflag &= ~(ECHO);
    term.c_cc[VMIN] = 10;
    term.c_cc[VTIME] = 0;
    tcsetattr(0, TCSAFLUSH, &term);
    return true;
}
void ExitProgram(int sig){
    exit(0);
}
int main(int argc, const char*** argv) {
    char c;

    do{
        std::cout << "Enter \"1\" for Non-canonical mode or \"2\" for Canonical mode: ";
        std::cin >> c;
        switch(c){
            case '1':
                NonCanonical();
                break;
            case '2':
                Canonical();
                break;
            default:
                std::cout << "You enter wrong value! Try again" << std::endl;
        }
    }while(c != '1' && c != '2');
    atexit(ResetInputMode);
    signal(SIGINT,ExitProgram);
    signal(SIGTERM,ExitProgram);
    while(true) {
        read(0,&c,term.c_cc[VMIN]);
        write(1,&c,term.c_cc[VMIN]);
    }
    return 0;
}

```

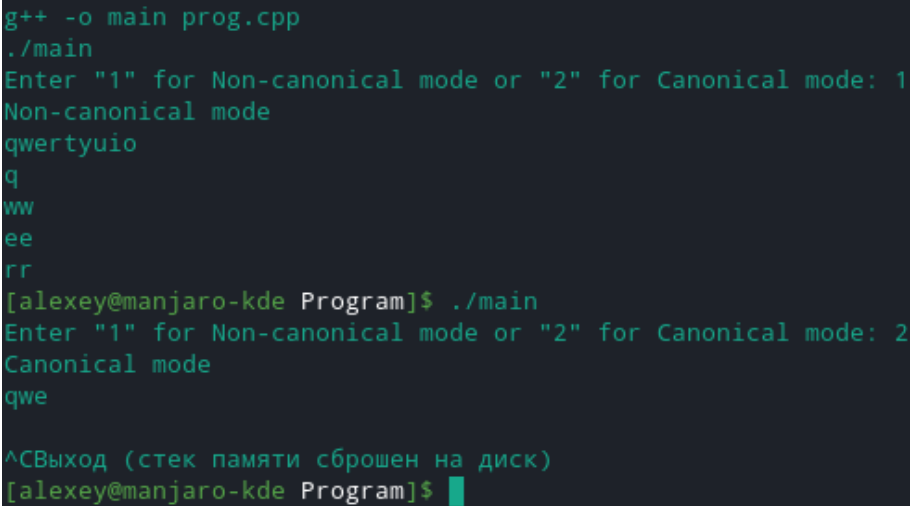
Результати виконання програми

На рис.1 продемонстровано роботу програми.

При першому виконанні програми використано неканонічний режим, при якому введення відбудеться лише після того, як у черзі введення буде не менше 10 байт. На скриншоті можна побачити, що у рядку “qwertyuio”, де кожна літера займає 1 байт, є потрібний розмір, тому рядок відображається весь. У наступних рядках при натиску на клавішу “Enter” у черзі не було потрібного об’єму даних, тому введенні символи не відображались до моменту заповнення черги.

При другому виконанні програми був обраний канонічний режим із заміненними сполученнями клавіш. У програмі, що продемонстрована на скриншоті було виконано наступний порядок дій:

1. Введення текстового рядку;
2. Використання символу EOF за заміненним сполученням клавіш (з Ctrl+D на Ctrl+A) — завершення вводу рядка;
3. Видалення поточного введенного рядка за допомогою керуючого символу KILL (заміна Ctrl+U на Ctrl+B);
4. Посилання сигналу для зупинки процесу за допомогою керуючого символу QUIT (заміна Ctrl+\ на Ctrl+C).



```
g++ -o main prog.cpp
./main
Enter "1" for Non-canonical mode or "2" for Canonical mode: 1
Non-canonical mode
qwertyuio
q
ww
ee
rr
[alexey@manjaro-kde Program]$ ./main
Enter "1" for Non-canonical mode or "2" for Canonical mode: 2
Canonical mode
qwe

^CВыход (стек памяти сброшен на диск)
[alexey@manjaro-kde Program]$
```

Рис.1 Результат виконання програми

Висновки

У ході виконання лабораторної роботи набули навичок управління терміналом та освоєно роботу у неканонічному режимі терміналу.

При виконанні індивідуального завдання було використано локальні режими терміналу ICANON та ECHO, була проведена робота з символами редагування рядків VEOF, VKILL та символом відправки сигналу VQUIT.

Реалізовано читування та запис введенного користувачем рядку символів (т.к. для роботи читування потрібно мати у черзі не менше 10 байт) без затримок у часі, відповідно до індивідуального завдання.