Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных технологий

Кафедра Информатики

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

Преподаватель \_\_\_\_\_\_\_\_ Пересунько П. В.

подпись, дата инициалы, фамилия

Студент КИ15–17Б \_\_\_\_\_\_\_\_\_\_\_ Радионов Т.В.

номер группы подпись, дата инициалы, фамилия

Красноярск 2016

**1 Цель работы**

Изучение особенностей программной реализации многозадачных приложений в ОС GNU/Linux.

**2 Исходные тексты программ**

#include <pthread.h>

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <semaphore.h>

#define W 3 // количество Писателей

#define R 3 // количество Читателей

unsigned int iter; // итерация

sem\_t sem\_access,sem\_readers,sem\_order; // семафоры доступа, читателей, порядка

unsigned int readers = 0; // количество читателей с доступом к ресурсу

void \*reader(void \*prm)

{

int num1=\*(int\*)prm;

int i=0,time;

for(i;i<iter;i++)

{

if (sem\_wait(&sem\_order)==0) printf("%d Читатель %d в очереди\n",i,num1); // cтавим читателя в очередь

sem\_wait(&sem\_readers); // запрашиваем читателей для управления их количеством

if (readers == 0) // если нет других читателей...

sem\_wait(&sem\_access); // ...запрашиваем доступ к ресурсу

readers++; // увеличиваем количество читателей ресурса

sem\_post(&sem\_order); // порядок прибытия семафора (мы зарезервировали)

sem\_post(&sem\_readers); // мы дали доступ читателям на данный момент

printf("%d Работает читатель %d\n",i,num1); // читатель может считать ресурс

time=1+rand()%10; // назначаем случайное время чтения ресурса

sleep(time); // читатель в процессе чтения

sem\_wait(&sem\_readers); // запрашиваем читателей для управления их количеством

readers--; // уменьшаем количество читателей ресурса

if (readers == 0) // если нет больше читателей ресурса...

sem\_post(&sem\_access); // ...даем доступ к ресурсу

sem\_post(&sem\_readers); // мы дали доступ читателям на данный момент

}

}

void \*writer(void \*prm)

{

int num2=\*(int\*)prm;

int j=0,time;

for(j;j<iter;j++)

{

if(sem\_wait(&sem\_order)==0) printf("%d Писатель %d в очереди\n",j,num2); // ставим писателя в очередь

sem\_wait(&sem\_access); // запрос доступа к ресурсу

sem\_post(&sem\_order); // порядок прибытия семафора (мы зарезервировали)

printf("%d Работает писатель %d\n",j,num2); // Here the writer can modify the resource at will

time=1+rand()%10; // назначаем случайное время записи из ресурса писателями

sleep(time); // писатель в процессе записи

sem\_post(&sem\_access); // Release exclusive access to the resource

}

}

void main()

{

pthread\_t threadRE[R]; // поток читателей

pthread\_t threadWR[W]; // поток писателей

sem\_init(&sem\_access,0,1); // семафор доступа

sem\_init(&sem\_readers,0,1);

sem\_init(&sem\_order,0,1);

printf("Введите количество проходов (итераций): ");

scanf("%d",&iter);

printf("Iter ОЧЕРЕДЬ/ВЫПОЛНЕНИЕ\n");

int i;

// Создаем потоки Писателей

for(i=0;i<W;i++)

{

pthread\_create(&(threadWR[i]),NULL,writer,(void\*)&i);

}

// Создаем потоки Читателей

for(i=0;i<R;i++)

{

pthread\_create(&(threadRE[i]),NULL,reader,(void\*)&i);

}

// Выделение потоков для писателей и читателей

for(i=0;i<R;i++)

{

pthread\_join(threadRE[i],NULL); // поток i читателя

}

for(i=0;i<W;i++)

{

pthread\_join(threadWR[i],NULL); // поток i писателя

}

// Уничтожаем семафоры

sem\_destroy(&sem\_access);

sem\_destroy(&sem\_readers);

sem\_destroy(&sem\_order);

}

**3 Примеры работы программ**

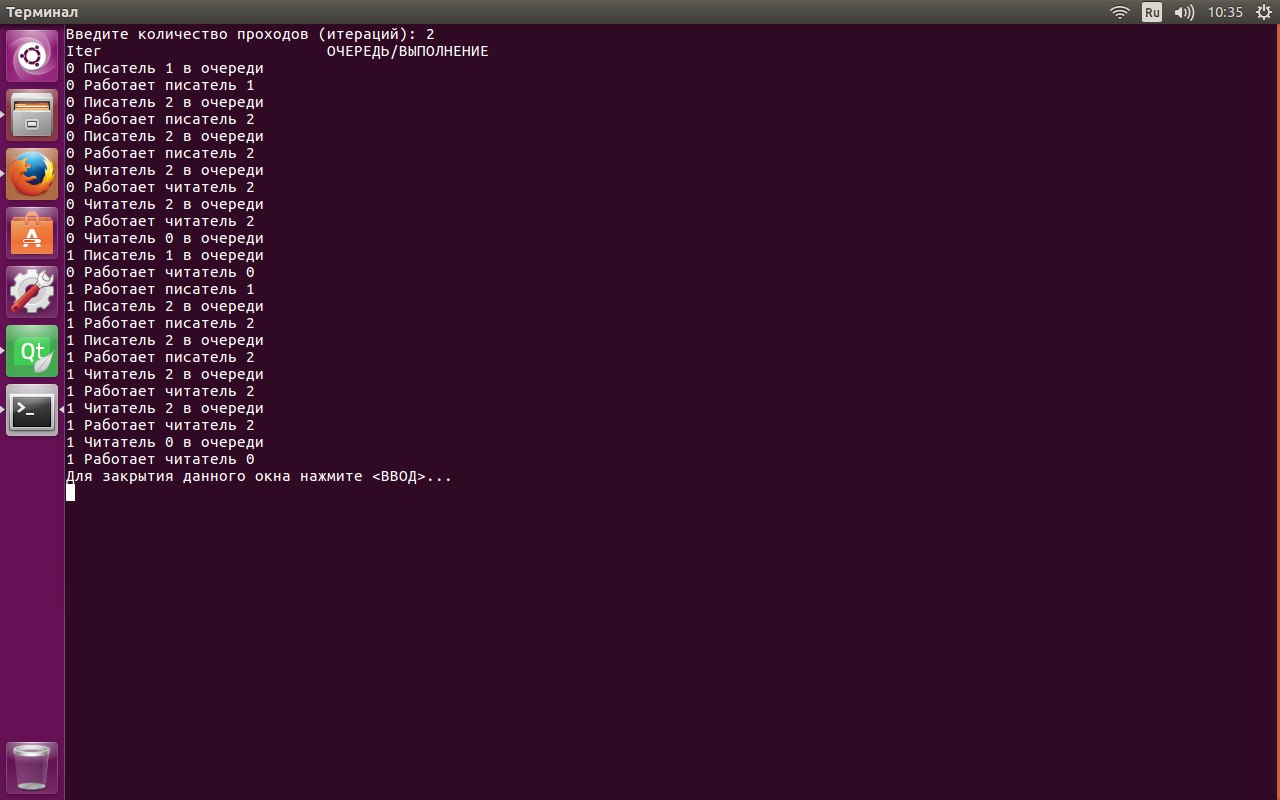


Рисунок 1 – Пример работы программы