Листинг:

#include <stdio.h>

#include <sys/msg.h>

#include <stdlib.h>

#define MAXLEN 80

int sieve\_of\_eratosthenes(int \*\*primes, const int min, const int max) {

int \*isPrime = (int \*)malloc(sizeof(int) \* (max + 1));

int i;

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

isPrime[i] = 1;

isPrime[0] = 0;

isPrime[1] = 0;

int count = 0;

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

{

if(isPrime[i] == 1)

{

++count;

int j;

for(j = 2 \* i; j <= max; j += i)

isPrime[j] = 0;

}

}

\*primes = (int \*)malloc(sizeof(int) \* count);

int j = 0;

for(i = min; i <= max; ++i)

{

if(isPrime[i] == 1)

{

(\*primes)[j++] = i;

}

}

return count;

}

struct msgbuf

{

long mtype;

char mtext[MAXLEN];

};

void send\_message(int qid, struct msgbuf \*qbuf, long type, char \*text){

qbuf->mtype = type;

strcpy(qbuf->mtext, text);

if((msgsnd(qid, (struct msgbuf \*)qbuf,

strlen(qbuf->mtext)+1, 0)) ==-1){

perror("msgsnd");

exit(1);

}

}

void read\_message(int qid, struct msgbuf \*qbuf, long type){

qbuf->mtype = type;

msgrcv(qid, (struct msgbuf \*)qbuf, MAXLEN, type, 0);

printf("Type: %ld Text: %s\n", qbuf->mtype, qbuf->mtext);

}

int main(int argc, char \*\*argv)

{

struct msgbuf buf;

struct msqid\_ds info;

int msqid, n;

char str1[] = "";

char str2[] = "";

printf(" СТАРТ, ПРОЦЕСС C ID %d\n\n", getpid());

msqid = msgget(IPC\_PRIVATE, 0600|IPC\_CREAT);

msgsnd(msqid, &buf, 1, 0);

msgctl(msqid, IPC\_STAT, &info);

printf("read-write: %o, cbytes = %lu, qnum = %lu, qbytes =%lu\n",

(ulong)info.msg\_perm.mode, (ulong) info.msg\_cbytes,

(ulong) info.msg\_qnum, (ulong) info.msg\_qbytes);

n = msgrcv(msqid, &buf, 1, buf.mtype, IPC\_NOWAIT);

printf("read %d bytes, type=%ld : %c\n\n",n,buf.mtype,buf.mtext[0]);

int \*primes = NULL;

int size, size2;

for (int i = 1; i < 3; i++){

sleep (rand()%4);

if (fork() == 0)

{

if(i==1)

{

printf("<-- ПЕРВЫЙ ПОДСЧЕТ В ИНТЕРВАЛЕ ОТ 10 ДО 40 -->\n", i, getpid());

int size = sieve\_of\_eratosthenes(&primes,10, 40);

char text[10];

buf.mtext[MAXLEN] = text;

buf.mtype = MAXLEN;

msqid = msgget(IPC\_PRIVATE, 0600|IPC\_CREAT);

sprintf(text, "%d", size);

send\_message(msqid, (struct msgbuf \*)&buf,

buf.mtype, text);

read\_message(msqid, &buf, buf.mtype);

printf("read %d bytes, type=%ld : %c\n",n,buf.mtype,buf.mtext[0]);

printf("--> сообщение : %s\n", buf.mtext);

printf("-- ГОТОВО -- %d НАЙДЕНО \n\n", size, getppid());

}

else

{

printf("<-- ВТОРОЙ ПОДСЧЕТ В ИНТЕРВАЛЕ ОТ 50 ДО 90 -->\n", i, getpid());

int size2 = sieve\_of\_eratosthenes(&primes,50, 90);

char text[10];

buf.mtext[MAXLEN] = text;

buf.mtype = MAXLEN;

msqid = msgget(IPC\_PRIVATE, 0600|IPC\_CREAT);

sprintf(text, "%d", size2);

send\_message(msqid, (struct msgbuf \*)&buf,

buf.mtype, text);

read\_message(msqid, &buf, buf.mtype);

printf("read %d bytes, type=%ld : %c\n",n,buf.mtype,buf.mtext[0]);

printf("--> сообщение : %s\n", buf.mtext);

printf("-- ГОТОВО -- %d НАЙДЕНО \n\n", size2, getppid());

}

exit(0);

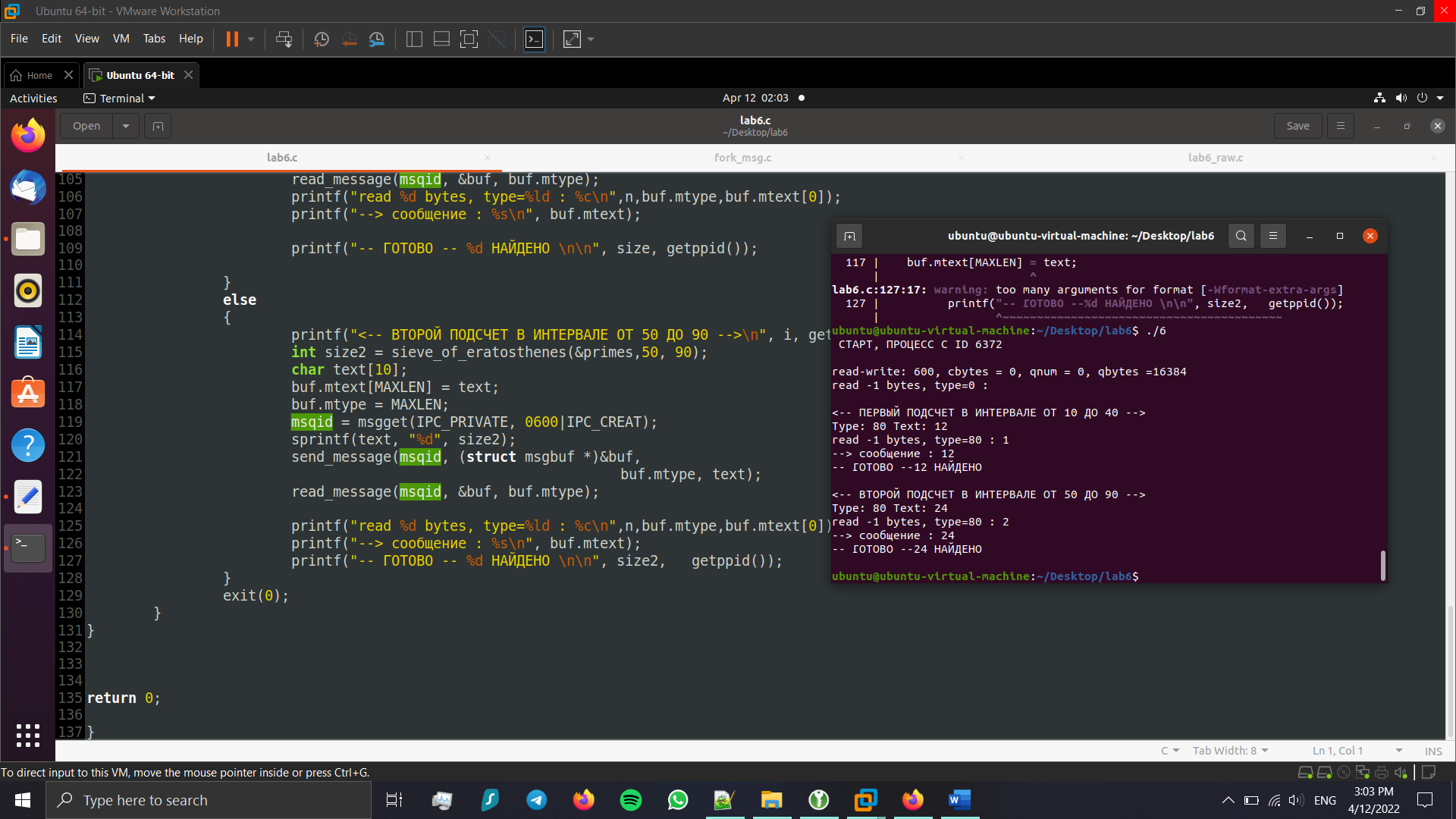
}

}

return 0;

}

Результат:



Описание:

Программа записывает в очередь сообщения о найденных простых числах из двух разных интервалов, поиск простых чисел производится в двух порожденных процессах.