**Московский Авиационный Институт**

**(Национальный Исследовательский Университет)**

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №2 по курсу**

**«Операционные системы»**

Группа: М80-203Б-22

Студент: Касумова Н.Р.

Преподаватель: Миронов Е.С.

Оценка:

Подпись:

Дата: 10.11.23

**Цель работы:**

Приобретение практических навыков в:

* Управление потоками в ОС
* Обеспечение синхронизации между потоками

**Постановка задачи**

Вариант 18

Найти образец в строке наивным алгоритмом.

Для выполнения второй лабораторной работы напишем основную функцию, то есть поиск подстроки наивным алгоритмом. Суть заключается в поиске всех допустимых сдвигов с помощью цикла, где string, pattern – строка и подстрока, а n , m – их длины соответственно. То есть цикл будет иметь вид: for(int i=0; i<n-m;i++) с условием if(string[i…i+m-1] == pattern).

Для работы с потоками я использую библиотеку <pthreads.h>, а также воспользуемся системными вызовами:

* pthread\_create(pthread, NULL(атрибуты функции), void, NULL(аргументы функции)); - создание потока. Возвращает тип int и имеет 4 аргумента (указатель на поток, атрибуты, указатель на функцию, аргументы функции)
* pthread\_join(pthread, NULL); - ожидание завершения процесса. Возвращает тип int и имеет два аргумента (указатель на поток, указатель на указатель в качестве аргумента для хранения возвращаемого значения

Сборка и запуск:

gcc lab2.c -pthread -o main

./main <number\_threads>

**Код программы**

main.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <errno.h>

#include <pthread.h>

#include <stdint.h>

#define MAX\_THREADS 1000

#define MAX\_STRING\_LENGTH 1000

char main\_string[MAX\_STRING\_LENGTH];

char substring[MAX\_STRING\_LENGTH];

int size, number\_threads;

int len\_substring, len\_string;

int \*result;

//int result\_size = 0;

void \*search\_substring(void \*arg) {

long thread\_id = (long)arg;

int start = thread\_id \* (len\_string / number\_threads);

int end = (thread\_id + 1) \* (len\_string / number\_threads);

//int j;

for(int i = start; i < end; i++) {

int j;

for(int j = 0; j < len\_substring; j++) {

if(main\_string[i + j] != substring[j]) {

break;

}

}

if (j == len\_substring) {

result[i] = 1;

//result\_size++;

}

}

pthread\_exit(NULL);

}

int main(int argc, char \*argv[]) {

if(argc != 2) {

perror("Неверное число параметров");

exit(-1);

}

number\_threads = atoi(argv[1]);

if (number\_threads <= 0 || number\_threads > MAX\_THREADS) {

perror("Недопустимое число потоков");

exit(-1);

}

printf("Введите строку: ");

fgets(main\_string, MAX\_STRING\_LENGTH, stdin);

len\_string = strlen(main\_string);

if(main\_string[len\_string - 1] == '\n') {

main\_string[len\_string - 1] = '\0';

len\_string--;

}

printf("Введите подстроку для поиска: ");

fgets(substring, MAX\_STRING\_LENGTH, stdin);

len\_substring = strlen(substring);

if(substring[len\_substring - 1] == '\n') {

substring[len\_substring - 1] = '\0';

len\_substring--;

}

result = (int \*)malloc(number\_threads \* sizeof(int));

pthread\_t threads[number\_threads];

int new\_thread;

for(int t = 0; t < number\_threads; t++) {

new\_thread = pthread\_create(&threads[t], NULL, search\_substring, (void \*)(intptr\_t)t);

if(new\_thread) {

perror("Ошибка при создании потока");

exit(-1);

}

}

for(int t = 0; t < number\_threads; t++) {

pthread\_join(threads[t], NULL);

}

int ans = 0;

for(int i = 0; i < number\_threads; i++) {

if(result[i] == 1) {

ans = 1;

break;

}

}

if(ans) {

printf("Подстрока найдена\n");

}

else {

printf("Подстрока не найдена\n");

}

free(result);

return 0;

}

**Strace**

ksu@DESKTOP-4J8FFR6:~/naidalaba2$ strace ./main abc fabcrakqwertyfenty 3

execve("./main", ["./main", "abc", "fabcrakqwertyfenty", "3"], 0x7ffda117f628 /\* 22 vars \*/) = 0

brk(NULL) = 0x55e12d97c000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fff206c4720) = -1 EINVAL (Invalid argument)

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=57411, ...}) = 0

mmap(NULL, 57411, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fb05f587000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libpthread.so.0", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220q\0\0\0\0\0\0"..., 832) = 832

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\365\267\271\263\357\340\36\367\256\306\221\334\217N',"..., 68, 824) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=157224, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fb05f585000

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\365\267\271\263\357\340\36\367\256\306\221\334\217N',"..., 68, 824) = 68

mmap(NULL, 140408, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fb05f562000

mmap(0x7fb05f568000, 69632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x6000) = 0x7fb05f568000

mmap(0x7fb05f579000, 24576, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17000) = 0x7fb05f579000

mmap(0x7fb05f57f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1c000) = 0x7fb05f57f000

mmap(0x7fb05f581000, 13432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fb05f581000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\346x\376T\245\322\302\t/\216G\353\v3\20^"..., 68, 880) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029592, ...}) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\346x\376T\245\322\302\t/\216G\353\v3\20^"..., 68, 880) = 68

mmap(NULL, 2037344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fb05f370000

mmap(0x7fb05f392000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7fb05f392000

mmap(0x7fb05f50a000, 319488, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19a000) = 0x7fb05f50a000

mmap(0x7fb05f558000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7fb05f558000

mmap(0x7fb05f55e000, 13920, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fb05f55e000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fb05f36d000

arch\_prctl(ARCH\_SET\_FS, 0x7fb05f36d740) = 0

mprotect(0x7fb05f558000, 16384, PROT\_READ) = 0

mprotect(0x7fb05f57f000, 4096, PROT\_READ) = 0

mprotect(0x55e12bf7e000, 4096, PROT\_READ) = 0

mprotect(0x7fb05f5c3000, 4096, PROT\_READ) = 0

munmap(0x7fb05f587000, 57411) = 0

set\_tid\_address(0x7fb05f36da10) = 162

set\_robust\_list(0x7fb05f36da20, 24) = 0

rt\_sigaction(SIGRTMIN, {sa\_handler=0x7fb05f568bf0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7fb05f576420}, NULL, 8) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7fb05f568c90, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7fb05f576420}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

brk(NULL) = 0x55e12d97c000

brk(0x55e12d99d000) = 0x55e12d99d000

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fb05eb6c000

mprotect(0x7fb05eb6d000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7fb05f36bfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tidptr=0x7fb05f36c9d0, tls=0x7fb05f36c700, child\_tidptr=0x7fb05f36c9d0) = 163

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fb05e36b000

mprotect(0x7fb05e36c000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7fb05eb6afb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tidptr=0x7fb05eb6b9d0, tls=0x7fb05eb6b700, child\_tidptr=0x7fb05eb6b9d0) = 164

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fb05db6a000

mprotect(0x7fb05db6b000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7fb05e369fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tidptr=0x7fb05e36a9d0, tls=0x7fb05e36a700, child\_tidptr=0x7fb05e36a9d0) = 165

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

write(1, "substring found in string\n", 26substring found in string

) = 26

write(1, "\320\222\321\200\320\265\320\274\321\217 \320\262\321\213\320\277\320\276\320\273\320\275\320\265\320\275\320\270\321\217:"..., 38Время выполнения: 8965

) = 38

exit\_group(0) = ?

+++ exited with 0 +++

**Пример работы**

n@MacBook-Pro-N lab\_2 % ./main abc fabcrakqwertyfenty 3

substring found in string

Время выполнения: 8965

**Вывод**

В ходе выполнения лабораторной работы я узнала как работать с несколькими потоками: как их создавать, выполнять с ними действия, их синхронизировать и объединять результаты после выполнения. Мною были получены навыки разработки многопоточных программ с параллельной обработкой одних больших данных.