Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №4 по курсу**

**«Операционные системы»**

Группа: М8О-215Б-23

Студент: Тараскаев Д.М.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 01.11.24

Москва, 2024

**Постановка задачи**

Вариант 25

Реализовать функции нахождения НОД наивным методом и через алгоритм Евклида.

Реализовать функции перевода числа из 10 СС в 2 СС или 3 СС.

**Общий метод и алгоритм решения**

В программе использовались следующие системные вызовы:

* dlopen - загружает динамическую библиотеку во время выполнения программы.
* dlclose - выгружает динамическую библиотеку
* dlsym - получает адрес функции или переменной из загруженной динамической библиотеки.
* dlerror - возвращает строку с описанием последней ошибки, произошедшей при вызове dlopen, dlsym или dlclose.

Алгоритм Евклида применяется к паре положительных целых чисел и формирует новую пару, которая состоит из меньшего числа и остатка от деления большего числа на меньшее. Процесс повторяется, пока числа не станут равными. Найденное число и есть наибольший общий делитель исходной пары.

Наивный алгоритм же проверяет делимость двух чисел от 2 до минимального из этих двух чисел.

Перевод из 10 СС в X CC реализован нахождением остатка при делении на X и делении числа на X, до тех пор пока число не станет равным 0. Найденные остатки и есть ответ.

**Код программы**

**prog1.c**

#include <stdio.h>

#include <stdlib.h>

int GCF(int a, int b);

char\* translation(long x);

int main() {

while (1) {

printf("1 - GCF\n2 - Translate number\n3 - exit\n");

int n;

scanf("%d", &n);

switch (n) {

case 1:

printf("Введите числа a b\n");

int a, b;

scanf("%d", &a);

scanf("%d", &b);

printf("%d\n", GCF(a, b));

break;

case 2:

printf("Введите число\n");

long x;

scanf("%d", &x);

printf("%s\n", translation(x));

break;

case 3:

exit(0);

default:

break;

}

}

return 0;

}

**prog2.c**

#include <stdio.h>

#include <stdlib.h>

#include <dlfcn.h>

int (\*GCF)(int a, int b);

char\* (\*translation)(long x);

void load\_library(int mode) {

void \*handle;

char \*error;

if (mode == 1) {

handle = dlopen("./libprogv1.so", RTLD\_LAZY);

}

else if (mode == 2) {

handle = dlopen("./libprogv2.so", RTLD\_LAZY);

}

if (!handle) {

fprintf(stderr, "%s\n", dlerror());

exit(EXIT\_FAILURE);

}

dlerror();

\*(void \*\*) (&GCF) = dlsym(handle, "GCF");

if ((error = dlerror()) != NULL) {

fprintf(stderr, "%s\n", error);

exit(EXIT\_FAILURE);

}

dlerror();

\*(void \*\*) (&translation) = dlsym(handle, "translation");

if ((error = dlerror()) != NULL) {

fprintf(stderr, "%s\n", error);

exit(EXIT\_FAILURE);

}

}

int main() {

int mode = 1;

load\_library(mode);

while (1) {

printf("Текущая реализация: %dv\n0 - Переключить реализацию\n1 - GCF\n2 - Translate number\n3 - Exit\n", mode);

int n;

scanf("%d", &n);

switch (n) {

case 0:

mode = mode % 2 + 1;

load\_library(mode);

break;

case 1:

printf("Введите числа a b\n");

int a, b;

scanf("%d", &a);

scanf("%d", &b);

printf("%d\n", GCF(a, b));

break;

case 2:

printf("Введите число\n");

long x;

scanf("%ld", &x);

printf("%s\n", translation(x));

break;

case 3:

exit(0);

default:

printf("Неверный пункт\n");

break;

}

}

return 0;

}

**Реализация функций GCF**

Наивный алгоритм:

int GCF(int a, int b) {

int mn = a < b ? a : b;

int gcd = 1;

for (int i = 2; i <= mn; ++i) {

if (a % i == 0 && b % i == 0) {

gcd = i;

}

}

return gcd;

}

Алгоритм Евклида:

int GCF(int a, int b) {

while (a != 0 && b != 0) {

if (a > b) {

a = a % b;

}

else {

b = b % a;

}

}

return a + b;

}

**Реализация перевода из 10 СС в X СС**

#include <stdlib.h>

#include <string.h>

char\* translation(long x) {

char\* ans = (char\*)malloc(64 \* sizeof(char));

int i = 0;

if (x == 0) {

ans[i++] = '0';

}

else {

while (x > 0) {

ans[i++] = (x % X) + '0';

x = x / X;

}

}

ans[i] = '\0';

int len = strlen(ans);

for (int j = 0; j < len / 2; j++) {

char temp = ans[j];

ans[j] = ans[len - j - 1];

ans[len - j - 1] = temp;

}

return ans;

}

**Makefile**

v1:

gcc -o prog1 prog1.c -L. -lprogv1

v2:

gcc -o prog1 prog1.c -L. -lprogv2

prog2:

gcc -o prog2 prog2.c -ldl

libs:

gcc -fPIC -shared -o libprogv2.so GCF1.c translation2.c

gcc -fPIC -shared -o libprogv1.so GCF1.c translation1.c

clean:

-rm prog1

-rm prog2

-rm \*.so

**Протокол работы программы**

**Тестирование prog1**

$ make libs

$ make v1

$ ./prog1

1 - GCF

2 - Translate number

3 - exit

1

Введите числа a b

10 15

5

1 - GCF

2 - Translate number

3 - exit

2

Введите число

14

1110

1 - GCF

2 - Translate number

3 - exit

3

**Тестирование prog2**

[abs@archlinux src]$ ./prog2

Текущая реализация: 1v

0 - Переключить реализацию

1 - GCF

2 - Translate number

3 - Exit

1

Введите числа a b

14 16

2

Текущая реализация: 1v

0 - Переключить реализацию

1 - GCF

2 - Translate number

3 - Exit

2

Введите число

20

10100

Текущая реализация: 1v

0 - Переключить реализацию

1 - GCF

2 - Translate number

3 - Exit

0

Текущая реализация: 2v

0 - Переключить реализацию

1 - GCF

2 - Translate number

3 - Exit

2

Введите число

20

202

Текущая реализация: 2v

0 - Переключить реализацию

1 - GCF

2 - Translate number

3 - Exit

**strace prog1:**

$ strace ./prog1

execve("./prog1", ["./prog1"], 0x7ffcf53e8d60 /\* 51 vars \*/) = 0

brk(NULL) = 0x60bc51ad6000

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

openat(AT\_FDCWD, "./glibc-hwcaps/x86-64-v3/libprogv1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "./glibc-hwcaps/x86-64-v2/libprogv1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "./libprogv1.so", 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\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=15040, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x78495b133000

getcwd("/home/abs/projects/os\_labs/lab4/src", 128) = 36

mmap(NULL, 16416, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x78495b12e000

mmap(0x78495b12f000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x78495b12f000

mmap(0x78495b130000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x78495b130000

mmap(0x78495b131000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x78495b131000

close(3) = 0

openat(AT\_FDCWD, "./glibc-hwcaps/x86-64-v3/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "./glibc-hwcaps/x86-64-v2/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "./libc.so.6", O\_RDONLY|O\_CLOEXEC) = -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=121891, ...}) = 0

mmap(NULL, 121891, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x78495b110000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/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\340\_\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

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2014520, ...}) = 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

mmap(NULL, 2034616, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x78495af1f000

mmap(0x78495af43000, 1511424, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x24000) = 0x78495af43000

mmap(0x78495b0b4000, 319488, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x195000) = 0x78495b0b4000

mmap(0x78495b102000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e3000) = 0x78495b102000

mmap(0x78495b108000, 31672, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x78495b108000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x78495af1c000

arch\_prctl(ARCH\_SET\_FS, 0x78495af1c740) = 0

set\_tid\_address(0x78495af1ca10) = 16594

set\_robust\_list(0x78495af1ca20, 24) = 0

rseq(0x78495af1d060, 0x20, 0, 0x53053053) = 0

mprotect(0x78495b102000, 16384, PROT\_READ) = 0

mprotect(0x78495b131000, 4096, PROT\_READ) = 0

mprotect(0x60bc23133000, 4096, PROT\_READ) = 0

mprotect(0x78495b16f000, 8192, PROT\_READ) = 0

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

munmap(0x78495b110000, 121891) = 0

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

getrandom("\xcc\xa0\x17\x79\x0e\x5d\x4f\x6d", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x60bc51ad6000

brk(0x60bc51af7000) = 0x60bc51af7000

write(1, "1 - GCF\n", 81 - GCF

) = 8

write(1, "2 - Translate number\n", 212 - Translate number

) = 21

write(1, "3 - exit\n", 93 - exit

) = 9

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

read(0, 2

"2\n", 1024) = 2

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \321\207\320\270\321\201\320\273\320\276\n", 26Введите число

) = 26

read(0, 10

"10\n", 1024) = 3

write(1, "1010\n", 51010

) = 5

write(1, "1 - GCF\n2 - Translate number\n", 291 - GCF

2 - Translate number

) = 29

write(1, "3 - exit\n", 93 - exit

) = 9

read(0, 15

"15\n", 1024) = 3

write(1, "1 - GCF\n2 - Translate number\n", 291 - GCF

2 - Translate number

) = 29

write(1, "3 - exit\n", 93 - exit

) = 9

read(0, 3

"3\n", 1024) = 2

lseek(0, -1, SEEK\_CUR) = -1 ESPIPE (Illegal seek)

exit\_group(0) = ?

+++ exited with 0 +++

**strace prog2:**

$ strace ./prog2

execve("./prog2", ["./prog2"], 0x7ffeb47c0b58 /\* 51 vars \*/) = 0

brk(NULL) = 0x5a829db04000

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

openat(AT\_FDCWD, "./glibc-hwcaps/x86-64-v3/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "./glibc-hwcaps/x86-64-v2/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "./libc.so.6", O\_RDONLY|O\_CLOEXEC) = -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=121891, ...}) = 0

mmap(NULL, 121891, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x77624070e000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/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\340\_\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

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2014520, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x77624070c000

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

mmap(NULL, 2034616, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x77624051b000

mmap(0x77624053f000, 1511424, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x24000) = 0x77624053f000

mmap(0x7762406b0000, 319488, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x195000) = 0x7762406b0000

mmap(0x7762406fe000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e3000) = 0x7762406fe000

mmap(0x776240704000, 31672, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x776240704000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x776240518000

arch\_prctl(ARCH\_SET\_FS, 0x776240518740) = 0

set\_tid\_address(0x776240518a10) = 16729

set\_robust\_list(0x776240518a20, 24) = 0

rseq(0x776240519060, 0x20, 0, 0x53053053) = 0

mprotect(0x7762406fe000, 16384, PROT\_READ) = 0

mprotect(0x5a829a6d9000, 4096, PROT\_READ) = 0

mprotect(0x776240766000, 8192, PROT\_READ) = 0

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

munmap(0x77624070e000, 121891) = 0

getrandom("\x3d\x53\xd9\xa9\x9b\x86\x8e\x8c", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x5a829db04000

brk(0x5a829db25000) = 0x5a829db25000

openat(AT\_FDCWD, "./libprogv1.so", 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\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=15040, ...}) = 0

getcwd("/home/abs/projects/os\_labs/lab4/src", 128) = 36

mmap(NULL, 16416, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x776240727000

mmap(0x776240728000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x776240728000

mmap(0x776240729000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x776240729000

mmap(0x77624072a000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x77624072a000

close(3) = 0

mprotect(0x77624072a000, 4096, PROT\_READ) = 0

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

write(1, "\320\242\320\265\320\272\321\203\321\211\320\260\321\217 \321\200\320\265\320\260\320\273\320\270\320\267\320\260\321\206\320"..., 40Текущая реализация: 1v

) = 40

write(1, "0 - \320\237\320\265\321\200\320\265\320\272\320\273\321\216\321\207\320\270\321\202\321\214 \321\200\320\265\320"..., 480 - Переключить реализацию

) = 48

write(1, "1 - GCF\n", 81 - GCF

) = 8

write(1, "2 - Translate number\n", 212 - Translate number

) = 21

write(1, "3 - Exit\n", 93 - Exit

) = 9

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

read(0, 0

"0\n", 1024) = 2

openat(AT\_FDCWD, "./libprogv2.so", 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\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=15040, ...}) = 0

getcwd("/home/abs/projects/os\_labs/lab4/src", 128) = 36

mmap(NULL, 16416, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x776240722000

mmap(0x776240723000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x776240723000

mmap(0x776240724000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x776240724000

mmap(0x776240725000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x776240725000

close(3) = 0

mprotect(0x776240725000, 4096, PROT\_READ) = 0

write(1, "\320\242\320\265\320\272\321\203\321\211\320\260\321\217 \321\200\320\265\320\260\320\273\320\270\320\267\320\260\321\206\320"..., 126Текущая реализация: 2v

0 - Переключить реализацию

1 - GCF

2 - Translate number

3 - Exit

) = 126

read(0, 3

"3\n", 1024) = 2

lseek(0, -1, SEEK\_CUR) = -1 ESPIPE (Illegal seek)

exit\_group(0) = ?

+++ exited with 0 +++

**Вывод**

Программа успешно демонстрирует работу с динамическими библиотеками в ОС Linux как во время компиляции программы, так и во время ее выполнения.