МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

##### ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

## Семинар №4

### по дисциплине

### «Языки программирования»

##### ***Выполнил:*** Студент группы P3234 Баянов Равиль Динарович

#### Преподаватель:

##### Гурьянова Аглая Геннадьевна

Санкт-Петербург

2023

# Задание 1

Условие:

**Задание 1** Создайте файл hello.txt с текстом Hello, mmap!. Используя [заготовку](/programming-languages/cse-programming-languages-fall-2023/main/-/blob/master/seminar-4/hello_mmap.asm), отобразите его в память и выведите текст из него в стандартный поток вывода. Не забудьте вызвать munmap (его номер системного вызова 11) и закрыть файл (close, номер системного вызова 3) по завершению работы с файлом.

Решение:

; hello\_mmap.asm

%define O\_RDONLY 0

%define PROT\_READ 0x1

%define MAP\_PRIVATE 0x2

%define SYS\_WRITE 1

%define SYS\_OPEN 2

%define SYS\_MMAP 9

%define FD\_STDOUT 1

%define SYS\_CLOSE 3

%define SYS\_MUNMAP 11

section .data

; This is the file name. You are free to change it.

fname: db 'hello.txt', 0

section .text

global \_start

; use exit system call to shut down correctly

exit:

mov rax, 60

xor rdi, rdi

syscall

; These functions are used to print a null terminated string

; rdi holds a string pointer

print\_string:

push rdi

call string\_length

pop rsi

mov rdx, rax

mov rax, SYS\_WRITE

mov rdi, FD\_STDOUT

syscall

ret

string\_length:

xor rax, rax

.loop:

cmp byte [rdi+rax], 0

je .end

inc rax

jmp .loop

.end:

ret

; This function is used to print a substring with given length

; rdi holds a string pointer

; rsi holds a substring length

print\_substring:

mov rdx, rsi

mov rsi, rdi

mov rax, SYS\_WRITE

mov rdi, FD\_STDOUT

syscall

ret

\_start:

; Вызовите open и откройте fname в режиме read only.

mov rax, SYS\_OPEN

mov rdi, fname

mov rsi, O\_RDONLY ; Open file read only

mov rdx, 0 ; We are not creating a file

; so this argument has no meaning

syscall

; rax holds the opened file descriptor now

;mov r12, rax

; Вызовите mmap c правильными аргументами

; Дайте операционной системе самой выбрать, куда отобразить файл

; Размер области возьмите в размер страницы

; Область не должна быть общей для нескольких процессов

; и должна выделяться только для чтения.

mov r8, rax

mov rax, SYS\_MMAP

mov rdi, 0

mov rsi, 0x1000

mov rdx, PROT\_READ

mov r10, MAP\_PRIVATE

mov r9, 0

syscall

; с помощью print\_string теперь можно вывести его содержимое

mov rdi, rax

call print\_string

mov rdi, rsi

mov rsi, 0x1000

mov rax, SYS\_MUNMAP

syscall

mov rdi, r8

mov rax, SYS\_CLOSE

syscall

call exit

Работа программы:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# Задание 2

Условие:

**Задание 2** Прочитайте документацию по системному вызову fstat (номер 5). Вас будет интересовать поле st\_size типа off\_t структуры struct stat, которую функция fstat заполняет. Используйте его, чтобы корректно вычислить размер файла при выводе данных, и выведите их, используя функцию print\_substring (принимает на вход два аргумента: адрес начала строки и количество байт для вывода). Полученный размер файла используйте в вызовах mmap, munmap и print\_substring.

Решение:

; hello\_fstat.asm

%define O\_RDONLY 0

%define PROT\_READ 0x1

%define MAP\_PRIVATE 0x2

%define SYS\_WRITE 1

%define SYS\_OPEN 2

%define SYS\_MMAP 9

%define FD\_STDOUT 1

%define SYS\_MUNMAP 11

%define SYS\_FSTAT 5

%define SYS\_CLOSE 3

section .data

; This is the file name. You are free to change it.

fname: db 'hello.txt', 0

section .text

global \_start

; use exit system call to shut down correctly

exit:

mov rax, 60

xor rdi, rdi

syscall

; These functions are used to print a null terminated string

; rdi holds a string pointer

print\_string:

push rdi

call string\_length

pop rsi

mov rdx, rax

mov rax, SYS\_WRITE

mov rdi, FD\_STDOUT

syscall

ret

string\_length:

xor rax, rax

.loop:

cmp byte [rdi+rax], 0

je .end

inc rax

jmp .loop

.end:

ret

; This function is used to print a substring with given length

; rdi holds a string pointer

; rsi holds a substring length

print\_substring:

mov rdx, rsi

mov rsi, rdi

mov rax, SYS\_WRITE

mov rdi, FD\_STDOUT

syscall

ret

\_start:

push r12

push r13

; Вызовите open и откройте fname в режиме read only.

mov rax, SYS\_OPEN

mov rdi, fname

mov rsi, O\_RDONLY ; Open file read only

mov rdx, 0 ; We are not creating a file

; so this argument has no meaning

syscall

;sizeof(struct stat) = 144

;offsetof(struct stat, st\_size) = 48

;sizeof(off\_t) = 8

sub rsp, 144

mov r13, rsp

mov rdi, rax

mov rsi, rsp

mov rax, SYS\_FSTAT

syscall

mov r8, rdi

mov rdi, 0

mov rsi, [r13+48]

mov rdx, PROT\_READ

mov r10, MAP\_PRIVATE

mov r9, 0

mov rax, SYS\_MMAP

syscall

mov r12, rax

mov rdi, rax

mov rsi, [r13+48]

call print\_substring

mov rdi, r12

mov rsi, [r13+48]

mov rax, SYS\_MUNMAP

syscall

mov rdi, r8

mov rax, SYS\_CLOSE

syscall

add rsp, 144

pop r12

pop r13

call exit

Работа программы:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# Задание 3

Условие:

**Задание 3** В этих файлах не хватает нескольких строчек чтобы можно было взаимодействовать с кодом друг друга. Допишите файлы так, чтобы функции print\_string и hello вызывалась и проверьте результат. Подсказка: вспомните, что нужно, чтобы из одного файла с C-кодом вызвать код из другого файла.

Решение:

void world(void);

void print\_string(char\* str);

Работа программы:

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

# Задание 4

Условие:

**Задание 4** Объедините ассемблерный код для вывода содержимого файла с кодом на языке C. Пусть ваша программа будет просить пользователя ввести название файла, а затем выведет его содержимое в стандартный поток вывода используя код, написанный в начале сегодняшнего семинара (сделайте из него функцию print\_file, которая будет принимать имя файла первым аргументом). Не забудьте, что для корректной работы необходимо следовать соглашениям о вызовах и сохранить callee-saved регистры, которые вы используете, в начале своей функции print\_file. Для вывода сообщений (например “Please enter file name: “) используйте собственную реализацию print\_string из сегодняшнего семинара.

Решение:

hello.c

#include <stdio.h>

#include <string.h>

void print\_string(char\* str);

void print\_file(char\* name\_file);

int main()

{

char filename[128];

print\_string("Please enter you file name:\n");

if (fgets(filename, 128, stdin) != NULL) {

filename[strlen(filename) - 1] = '\0';

}

print\_file(filename);

return 0;

}

**lib.asm**

; hello\_mmap.asm

%define O\_RDONLY 0

%define PROT\_READ 0x1

%define MAP\_PRIVATE 0x2

%define SYS\_WRITE 1

%define SYS\_OPEN 2

%define SYS\_CLOSE 3

%define SYS\_FSTAT 5

%define SYS\_MMAP 9

%define SYS\_MUNMAP 11

%define FD\_STDOUT 1

section .data

; This is the file name. You are free to change it.

fname: db 'hello.txt', 0

section .text

global print\_file

global print\_string

; use exit system call to shut down correctly

exit:

mov rax, 60

xor rdi, rdi

syscall

; These functions are used to print a null terminated string

; rdi holds a string pointer

print\_string:

push rdi

call string\_length

pop rsi

mov rdx, rax

mov rax, SYS\_WRITE

mov rdi, FD\_STDOUT

syscall

ret

string\_length:

xor rax, rax

.loop:

cmp byte [rdi+rax], 0

je .end

inc rax

jmp .loop

.end:

ret

; This function is used to print a substring with given length

; rdi holds a string pointer

; rsi holds a substring length

print\_substring:

mov rdx, rsi

mov rsi, rdi

mov rax, SYS\_WRITE

mov rdi, FD\_STDOUT

syscall

ret

print\_file:

;rdi - название файла

push r12 ;fd

push r13 ;buf

push r14 ;addr

;open

mov rax, SYS\_OPEN

mov rsi, O\_RDONLY

mov rdx, 0

syscall

;save

mov r12, rax ;save fd

sub rsp, 144 ;reserve stack

mov r13, rsp ;save buffer

;fstat

mov rdi, r12 ;fd

mov rsi, r13 ;buffer

mov rax, SYS\_FSTAT ;syscall FSTAT

syscall

;mmap

mov rdi, 0 ;addr

mov rsi, [r13+48] ;length

mov rdx, PROT\_READ ;prot

mov r10, MAP\_PRIVATE ;flags

mov r8, r12 ;fd

mov r9, 0 ;offset

mov rax, SYS\_MMAP ;syscall MMAP

syscall

mov r14, rax ;save addr

;substring

mov rdi, r14 ;addr

mov rsi, [r13+48] ;length

call print\_substring

;munmap

mov rdi, r14 ;addr

mov rsi, [r13+48] ;length

mov rax, SYS\_MUNMAP ;syscall MUNMAP

syscall

;close

mov rdi, r12 ;fd

mov rax, SYS\_CLOSE ;syscall CLOSE

syscall

add rsp, 144

pop r14

pop r13

pop r12

call exit

Работа программы:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание