

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 Р3234

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](#), отобразите его в память и выведите текст из него в стандартный поток вывода. Не забудьте вызвать mmap (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 с 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:

```
call exit
ravvcheck@RavvCheck:~/seminar4$ ls -la
total 32
drwxr-xr-x 2 ravvcheck ravvcheck 4096 Oct 19 16:40 .
drwxr-x--- 9 ravvcheck ravvcheck 4096 Oct 19 16:39 ..
-rw-r--r-- 1 ravvcheck ravvcheck  13 Oct 19 16:39 hello.txt
-rwxr-xr-x 1 ravvcheck ravvcheck 9080 Oct 19 16:34 hello_mmap
-rw-r--r-- 1 ravvcheck ravvcheck 2261 Oct 19 16:34 hello_mmap.asm
-rw-r--r-- 1 ravvcheck ravvcheck 1216 Oct 19 16:34 hello_mmap.o
ravvcheck@RavvCheck:~/seminar4$ ./hello_mmap
Hello, mmap!
ravvcheck@RavvCheck:~/seminar4$ cat hello.txt
Hello, mmap!
ravvcheck@RavvCheck:~/seminar4$ █
```

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:

```
ravvcheck@RavvCheck:~/seminar4$ vim hello_fstat.asm
ravvcheck@RavvCheck:~/seminar4$ nasm -f elf64 -o hello_fstat.o hello_fstat.asm
ravvcheck@RavvCheck:~/seminar4$ ld -o hello_fstat hello_fstat.o
ravvcheck@RavvCheck:~/seminar4$ ./hello_fstat
Hello, mmap!
ravvcheck@RavvCheck:~/seminar4$ ls -la
total 52
drwxr-xr-x 2 ravvcheck ravvcheck 4096 Oct 19 17:06 .
drwxr-x--- 9 ravvcheck ravvcheck 4096 Oct 19 17:06 ..
-rw-r--r-- 1 ravvcheck ravvcheck  13 Oct 19 16:39 hello.txt
-rwxr-xr-x 1 ravvcheck ravvcheck 9080 Oct 19 17:06 hello_fstat
-rw-r--r-- 1 ravvcheck ravvcheck 1911 Oct 19 17:06 hello_fstat.asm
-rw-r--r-- 1 ravvcheck ravvcheck 1264 Oct 19 17:06 hello_fstat.o
-rwxr-xr-x 1 ravvcheck ravvcheck 9080 Oct 19 16:34 hello_mmap
-rw-r--r-- 1 ravvcheck ravvcheck 2261 Oct 19 16:34 hello_mmap.asm
-rw-r--r-- 1 ravvcheck ravvcheck 1216 Oct 19 16:34 hello_mmap.o
ravvcheck@RavvCheck:~/seminar4$
```

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 С-кодом вызвать код из другого файла.

Решение:

```
void world(void);
void print_string(char* str);
```

Работа программы:

```
ravvcheck@RavvCheck:~/seminar4/task3$ ls -la
total 52
drwxr-xr-x 2 ravvcheck ravvcheck 4096 Oct 19 17:30 .
drwxr-xr-x 3 ravvcheck ravvcheck 4096 Oct 19 17:17 ..
-rwxr-xr-x 1 ravvcheck ravvcheck 17576 Oct 19 17:30 hello
-rw-r--r-- 1 ravvcheck ravvcheck 93 Oct 19 17:25 hello.c
-rw-r--r-- 1 ravvcheck ravvcheck 3928 Oct 19 17:30 hello.o
-rw-r--r-- 1 ravvcheck ravvcheck 561 Oct 19 17:18 makefile
-rw-r--r-- 1 ravvcheck ravvcheck 516 Oct 19 17:25 string.asm
-rw-r--r-- 1 ravvcheck ravvcheck 49 Oct 19 17:30 string.inc
-rw-r--r-- 1 ravvcheck ravvcheck 2384 Oct 19 17:30 string.o
ravvcheck@RavvCheck:~/seminar4/task3$ cat string.inc
void world(void);
void print_string(char* str);

ravvcheck@RavvCheck:~/seminar4/task3$ cat hello.c
/* hello.c */

#include "string.inc"

int main() {
    print_string("hello");
    world();
}
ravvcheck@RavvCheck:~/seminar4/task3$ ./hello
hello, world!
ravvcheck@RavvCheck:~/seminar4/task3$
```


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:

```

ravvcheck@RavvCheck:~/seminar4/task4$ cat hello.txt
Hello, my friends!
ravvcheck@RavvCheck:~/seminar4/task4$ cat makefile
CFLAGS      = -g -O2 -Wall -Werror -std=c11 -Wdiscarded-qualifiers -Wincompatible-pointer-types -Wint-conversion
CC          = gcc
LD          = gcc
ASM         = nasm
ASMFLAGS    = -felf64 -g

# Если есть код на C, то компоновку тоже нужно производить
# с помощью gcc, а не ld
hello: hello.o lib.o
    $(LD) -o $@ $^

hello.o: hello.c
    $(CC) -c $(CFLAGS) -o $@ $<

lib.o: lib.asm
    $(ASM) $(ASMFLAGS) -o $@ $<

clean:
    $(RM) hello hello.o string.o

.PHONY: clean
ravvcheck@RavvCheck:~/seminar4/task4$ ./hello
Please enter you file name:
hello.txt
Hello, my friends!
ravvcheck@RavvCheck:~/seminar4/task4$ ./hello
Please enter you file name:
c
ravvcheck@RavvCheck:~/seminar4/task4$ █

```