Национальный исследовательский университет «МЭИ»

Институт информационных и вычислительных технологий

Кафедра Вычислительных технологий

Отчет по лабораторной работе № 5

по курсу «Операционные системы»

Группа А-06-20

Выполнили:

Дереберя Т.М.

Зинченко С.К.

Проверил:

Емельянов Д.М.

Github:

https://github.com/FunnyBananchik/OS\_Labs/tree/main/lab5

Москва

2023

## Постановка задания

Лабораторная работа №5 выполняется индивидуально или бригадами по 2 человека. При выполнении бригадой необходимо дополнительно подготовить документацию на разработанное ПО (с примерами работы, скриншотами и комментариями к коду).

1. Базовое решение для успешной сдачи лабораторной работы. Разработать Операционную систему на языке ассемблер под архитектуру х86 (достаточно одного загрузочного сектора), которая при загрузке осуществляет вывод собственного названия и лого в ASCII графике.
2. Для получения Отличной оценки также необходимо. Разработать Операционную систему на языке ассемблер под архитектуру х86 (достаточно одного загрузочного сектора), осуществляющую взаимодействие с пользователем. Например, ветвление диалога или проведение вычислений на основе вводимых с клавиатуры данных: y/n, имя, число и т.п.
3. Для получения дополнительных баллов в рамках выполнения ЛР можно дополнительно реализовать какую-либо игровую механику (карточные игры, двумерные игры, перемещение по лабиринтам, flappy bird и др., в обоснованно упрощенном варианте) или особое цветовое и стилевое оформление интерфейса.

Для отчетности необходимо:

* продемонстрировать работоспособность преподавателю, ответить на уточняющие вопросы по разработанному исходному коду
* предоставить в электронном виде документацию/отчет, если в бригаде более одного человека

## Код основного задания

org 0x7C00

mov si, logo

call print

mov si, hello

call print

call input\_name

mov si, hello\_name

call print

mov si, play

call print

call choose\_game

jmp $

print:

mov ah, 0x0e

lodsb

print\_loop:

int 0x10

lodsb

cmp al, 0x00

jne print\_loop

ret

input\_name:

mov bx, hello\_name + 7

input\_loop:

mov ah, 0x00

int 0x16

mov ah, 0x0e

int 0x10

cmp al, 8

je input\_back

cmp al, 13

je input\_end

mov [bx], al

inc bx

jmp input\_loop

input\_end:

mov byte [bx], 0

ret

input\_back:

dec bx

jmp input\_loop

choose\_game:

mov ah, 0x00

int 0x16

cmp al, 'y'

je print\_yes\_game

cmp al, 'n'

je print\_no\_game

jmp choose\_game

choose\_first\_game:

mov ah, 0x00

int 0x16

cmp al, 'y'

je print\_first\_win

cmp al, 'n'

je print\_first\_lose

jmp choose\_first\_game

print\_first\_win:

mov si, first\_win

call print

ret

print\_first\_lose:

mov si, first\_lose

call print

ret

print\_yes\_game:

mov si, first\_game

call print

call choose\_first\_game

ret

print\_no\_game:

mov si, no\_game

call print

ret

first\_win:

db 'Oh, it is 2. You win!', 0

first\_lose:

db 'I HAVE 20! YOU LOSE! MUHAHA', 0

first\_game:

db 'Lets start! You have 19. Once more? (y\n)', 10, 13, 0

no\_game:

db 'I dont either.', 0

logo:

db '\_\_ \_\_\_ |\\_\_\_/|', 10, 13, '| \ / ) (', 10, 13, '| | / =\ /=', 10, 13, '|\_/ /\_\_ )===(', 10, 13, ' \_ \_ / \', 10, 13, '/ \ / \ | \_ |', 10, 13, '| | \ / ( ( \', 10, 13, '| | \ \ ) ) /', 10, 13, '\\_/ \\_/ \\_( (\_/', 10, 13, 0

hello:

db 'Hello, user! Whats your name?', 10, 13, 0

play:

db '! Do you want to play blackjack? (y\n)', 10, 13, 0

hello\_name:

db 'Hello, '

times 510 - ($ - $$) db 0 ;

dw 0xaa55 ;

## Код и пояснения к игре

Была реализована упрощённая версия игры «Морской бой». Имеется игровое поле 4х4, на котором уже расположены корабли (один двухпалубный корабль и 3 однопалубных).

У игрока есть 10 попыток для того, чтобы найти все корабли. Для выбора клетки игрок должен ввести номер строки и номер столбца (от 1 до 4).

В случае попадания в этой клетке ставится «х», иначе «о».

Когда игрок находит все корабли, то выводится сообщение с поздравлениями. Если за 10 попыток найдены не все корабли, то выводится сообщение о проигрыше.

**Код:**

org 0x7C00

mov cx, 0x00

mov dx, 0x00

call main

jmp $

map:

db 1, 2, 8, 10, 16, 0

map2:

db '\_\_\_\_', 10, 13, '\_\_\_\_', 10, 13, '\_\_\_\_', 10, 13, '\_\_\_\_', 10, 13, 0

row:

db 'Enter row number:', 10, 13, 0

col:

db 10, 13, 'Enter column number:', 10, 13, 0

incor:

db 10, 13, 'Incorrect data!', 10, 13, 0

input:

mov ah, 0x00

int 0x16

mov ah, 0x0e

int 0x10

cmp al, '1'

jl err

cmp al, '4'

jg err

ret

err:

mov si, incor

call print

jmp input

main:

mov si, row

call print

call input

mov bl, 0

call korr

mov si, col

call print

call input

add al, bl

sub al, 0x30

call proverka

mov al, 10

int 0x10

mov al, 13

int 0x10

mov si, map2

call print

cmp cx, 0x05

je win

inc dx

cmp dx, 0x0a

je lose

jmp main

win\_txt:

db 'Congratulations, you win!^-^', 10, 13, 'The end.', 0

win:

mov si, win\_txt

call print

ret

lose\_txt:

db 'You lose:(', 10, 13, 'The end.', 0

lose:

mov si, lose\_txt

call print

ret

promah:

db 10, 13, 'Miss!', 0

proverka:

mov bx, map

preproverka:

cmp al, [bx]

je outmap

inc bx

cmp byte [bx], 0x00

jne preproverka

mov bx, map2 - 1

call check

call plus

mov byte [bx], 'o'

mov si, promah

call print

ret

check:

cmp al, 0x0d

jge addsix

cmp al, 0x09

jge addfour

cmp al, 0x05

jge addtwo

ret

addsix:

add al, 6

ret

addfour:

add al, 4

ret

addtwo:

add al, 2

ret

plus:

inc bx

dec al

cmp al, 0x00

jne plus

ret

popal:

db 10, 13, 'Hit!', 0

outmap:

inc cx

mov bx, map2 - 1

call check

call plus

mov byte [bx], 'x'

mov si, popal

call print

ret

korr:

cmp al, '1'

jne korr2

ret

korr2:

add bl, 4

dec al

jmp korr

print:

mov ah, 0x0e

lodsb

printloop:

int 0x10

lodsb

cmp al, 0x00

jne printloop

ret

times 510-($-$$) db 0

dw 0xAA55

## Тестирование основной программы

1. Программа запустилась и вывела логотип и вопрос для пользователя.

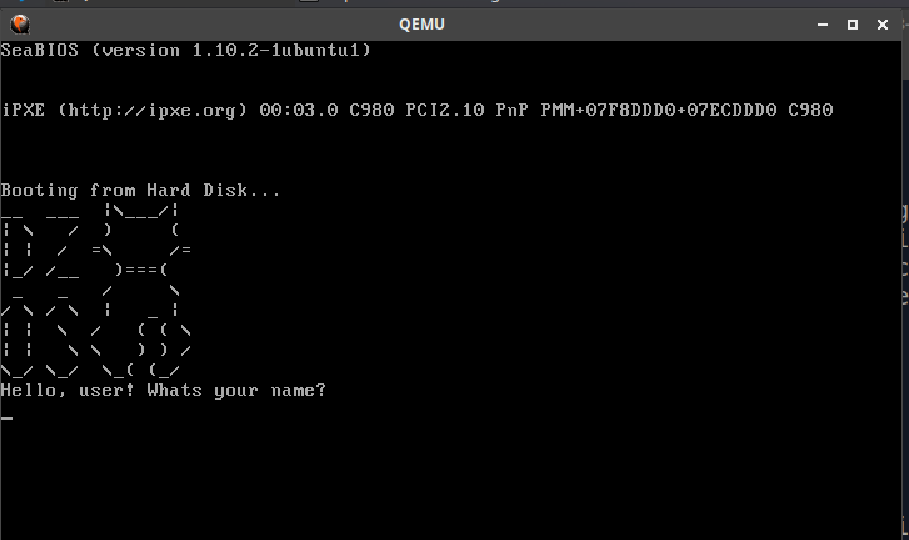


Рис.1 – Запуск программы

1. Вводилось имя «Tanya» и «n».

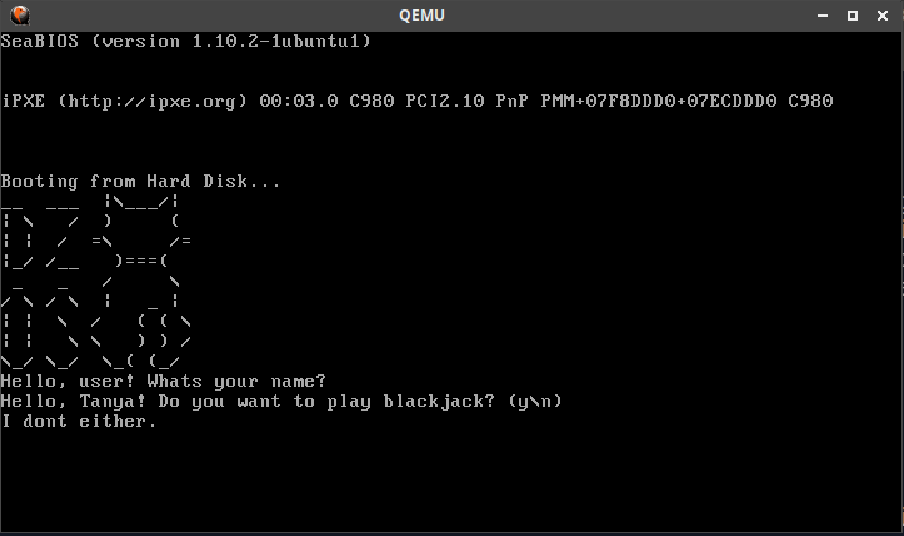


Рис.2 – Взаимодействие с пользователем

1. На вопрос, хочет ли пользователь поиграть, отвечаем «y». На следующий вопрос тоже отвечаем «y».

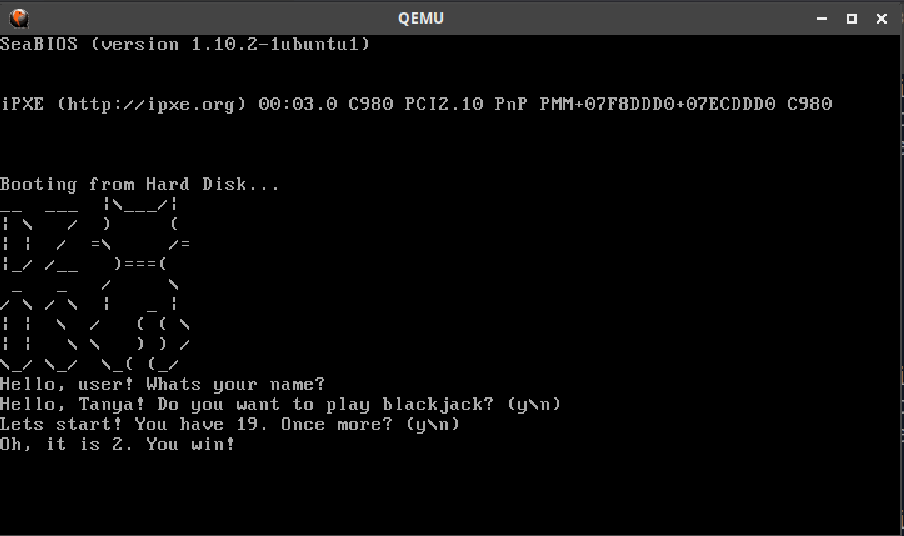


Рис.3 – Победа пользователя в игре

1. На вопрос, хочет ли пользователь поиграть, отвечаем «y». На следующий вопрос отвечаем «n».

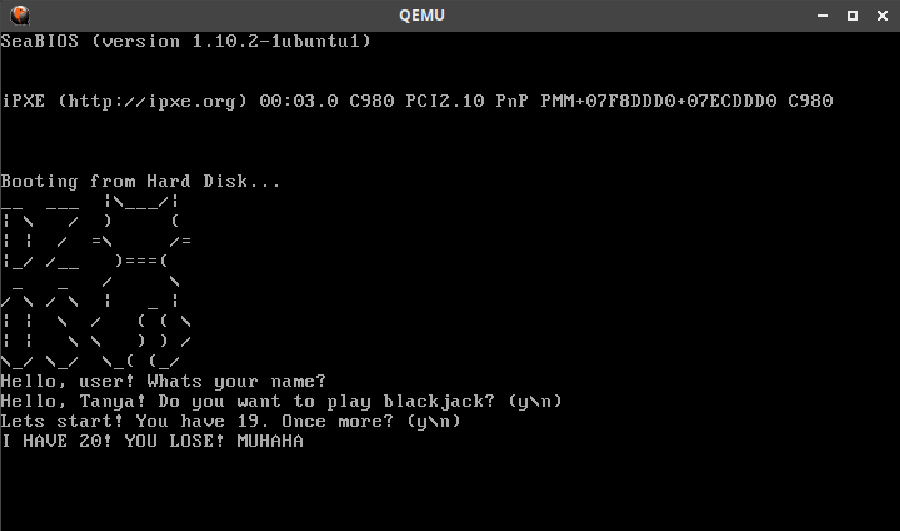


Рис.4 – Проигрыш пользователя в игре

## Тестирование игры

1. Первый ход в игре и попадание.

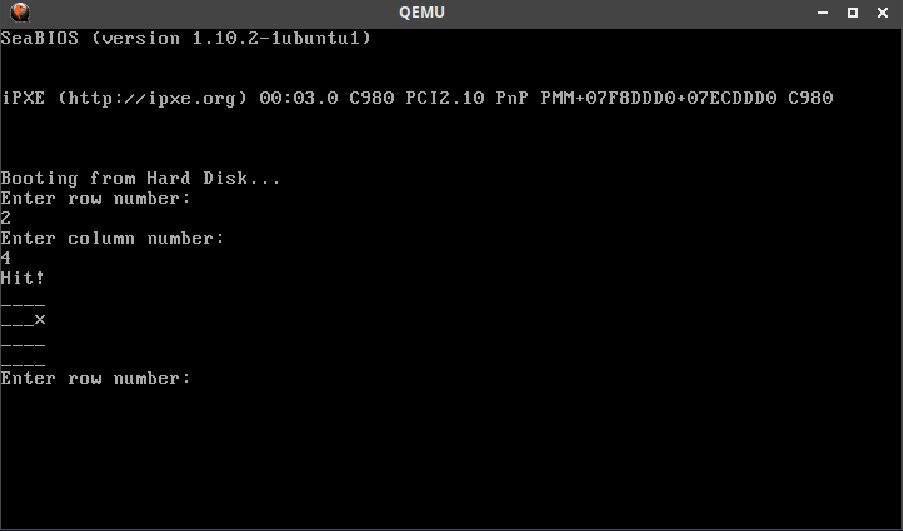


Рис.5 – Первый ход и попадание в игре

1. Игрок не попадает.

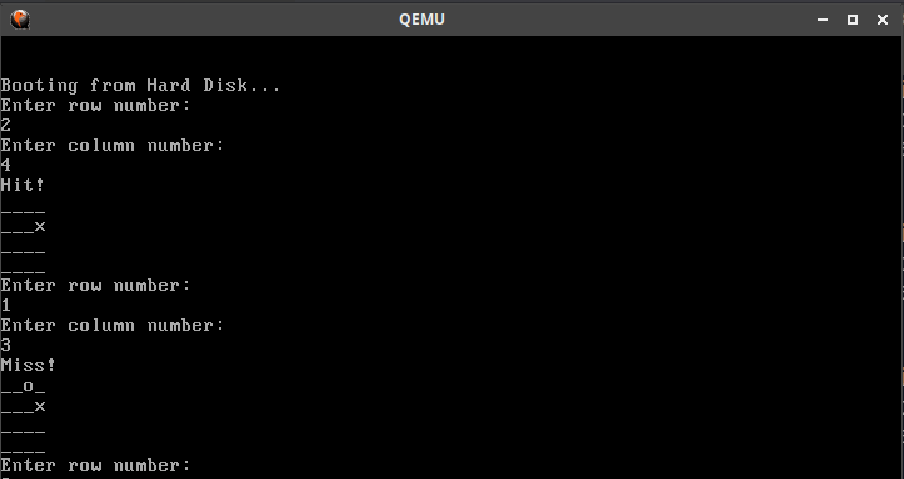


Рис.6 – Промах игрока

1. Игрок выигрывает.

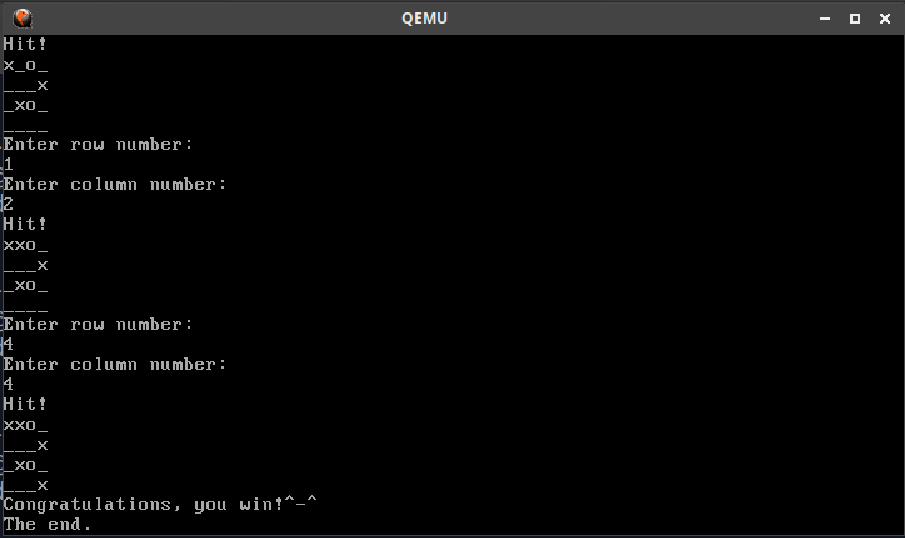


Рис.6 – Победа игрока

1. Игрок проигрывает.

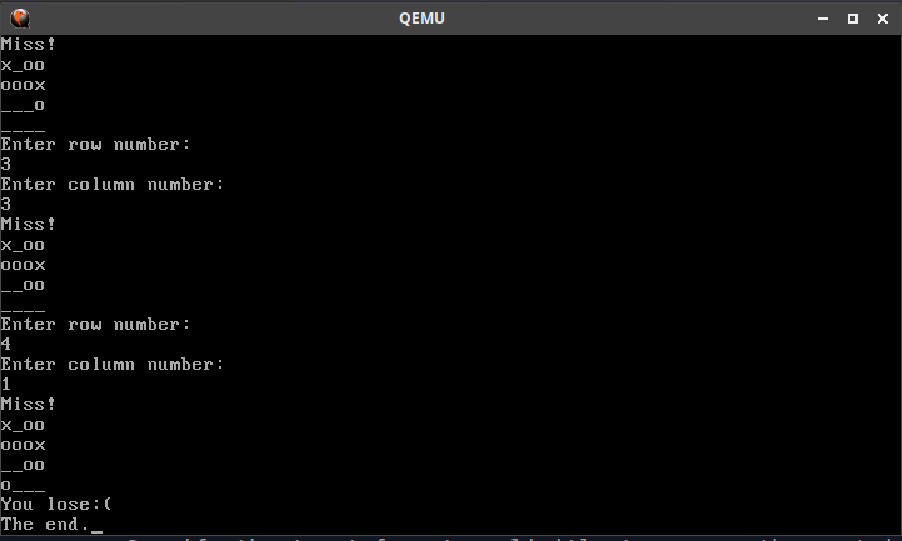


Рис.7 – Проигрыш игрока

1. Игрок вводит некорректные данные.

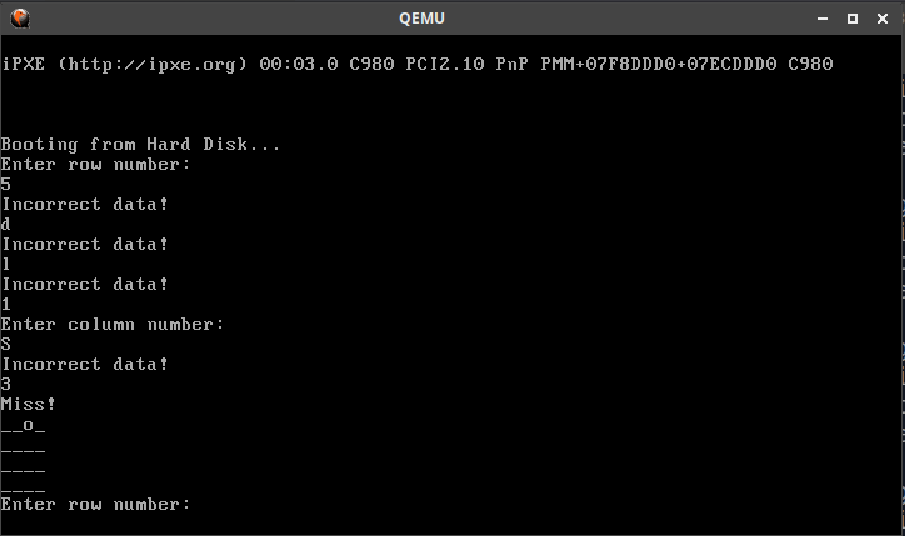


Рис.7 – Ввод некорректных данных