Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Архитектура вычислительных систем»

**ОТЧЕТ**

к практической работе №3

на тему:

**«МУЛЬТИЗАДАЧНОСТЬ В ЗАЩИЩЕННОМ РЕЖИМЕ»**

БГУИР 6-05-0612-02 005

|  |
| --- |
| Выполнила студентка группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  КАЛИНОВСКАЯ Анастасия Александровна |
|  |
| (дата, подпись преподавателя) |

Минск 2025

# 1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Написать программу, реализующую мультизадачность в защищенном режиме. Программа должна переключить процессор в защищенный режим, а затем запустить на выполнение 2-3 задачи.

# 2 ВЫПОЛНЕНИЕ РАБОТЫ

В рамках задания была разработана загрузочная программа и ядро, реализующие переключение процессора в защищённый режим и запуск двух задач в псевдомультизадачном режиме. После загрузки ядра с помощью BIOS-прерывания int 13h, происходит инициализация таблиц GDT и IDT, установка сегментных регистров и активация защищённого режима через флаг CR0. В защищённом режиме по нажатию клавиши запускается цикл переключения между двумя задачами, каждая из которых выводит сообщение на экран с задержкой. Переключение реализовано вручную через последовательные переходы между задачами, имитируя многозадачность. Программа протестирована в QEMU и написана на языке ассемблера NASM.

Листинг файла boot.bin.

[BITS 16]

[ORG 0x7C00]

start:

mov ax, 0x0000

mov es, ax

mov bx, 0x8000 ; загрузим ядро в 0x0000:0x8000 = 0x8000

mov ah, 0x02 ; BIOS: read sectors

mov al, 5 ; 1 сектор

mov ch, 0 ; цилиндр

mov cl, 2 ; сектор 2 (сектор 1 — загрузчик)

mov dh, 0 ; головка

mov dl, 0x80 ; диск (0 = A:)

int 13h

jc disk\_error

jmp 0x0000:0x8000 ; перейти к ядру

disk\_error:

mov si, disk\_msg

call print\_bios

jmp $

print\_bios:

mov ah, 0x0E

.next:

lodsb

cmp al, 0

je .done

int 10h

jmp .next

.done:

ret

disk\_msg db "Disk read error!", 0

times 510-($-$$) db 0

dw 0xAA55

Листинг файла kernel.bin.

[BITS 16]

[ORG 0x8000]

jmp start

; ---------------- GDT ----------------

gdt\_start:

gdt\_null: dq 0

gdt\_code: dw 0xFFFF, 0x0000, 0x9A00, 0x00CF

gdt\_data: dw 0xFFFF, 0x0000, 0x9200, 0x00CF

gdt\_end:

gdt\_descriptor:

dw gdt\_end - gdt\_start - 1

dd gdt\_start

; ---------------- IDT ----------------

idt\_start:

times 256 dq 0

idt\_end:

idt\_descriptor:

dw idt\_end - idt\_start - 1

dd idt\_start

; ---------------- Real Mode Entry ----------------

start:

mov si, msg\_rm

call print\_bios

.wait\_rm:

mov ah, 0

int 16h

cmp al, 'P'

je enter\_pm

jmp .wait\_rm

enter\_pm:

cli

lgdt [gdt\_descriptor]

lidt [idt\_descriptor]

mov eax, cr0

or eax, 1

mov cr0, eax

jmp 0x08:pm\_entry

; ---------------- Protected Mode ----------------

[BITS 32]

pm\_entry:

mov ax, 0x10

mov ds, ax

mov es, ax

mov ss, ax

mov esp, 0x90000

mov esi, msg\_pm

mov edi, 0xB8000

call print\_pm

.wait\_pm:

call read\_key\_pm

cmp al, 0x1F ; 'S'

jne .wait\_pm

jmp task1 ; запуск задачи 1

read\_key\_pm:

.wait\_press:

in al, 0x64

test al, 1

jz .wait\_press

in al, 0x60

cmp al, 0xF0

je .wait\_press

ret

; ---------------- Задачи ----------------

task1:

mov byte [color], 0x01

mov esi, msg\_task1

mov edi, 0xB8000 + 160

call print\_pm

call delay

jmp task2

task2:

mov byte [color], 0x07

mov esi, msg\_task2

mov edi, 0xB8000 + 160

call print\_pm

call delay

jmp task1

; ---------------- Задержка ----------------

delay:

mov ecx, 50000000

.loop:

dec ecx

jnz .loop

ret

; ---------------- Вывод ----------------

print\_bios:

mov ah, 0x0E

.next:

lodsb

cmp al, 0

je .done

int 10h

jmp .next

.done:

ret

print\_pm:

.next:

lodsb

cmp al, 0

je .done

mov ah, [color]

stosw

jmp .next

.done:

ret

; ---------------- Сообщения ----------------

msg\_rm db "Real Mode. Press P to enter Protected Mode.", 13, 10, 0

msg\_pm db "Protected Mode. Press S to start tasks.", 0

msg\_task1 db "Task1 running.", 0

msg\_task2 db "Task2 running.", 0

color db 0x07

# 3 АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

После перехода в защищённый режим программа успешно запускает две задачи, которые поочерёдно выводят сообщения на экран. На первом изображении отображается сообщение *«Task1 running.»* с цветом, заданным для первой задачи, что подтверждает корректную инициализацию видеопамяти и выполнение кода задачи 1. На втором изображении видно сообщение *«Task2 running.»* с другим цветом, соответствующим второй задаче, что демонстрирует успешное переключение между задачами.

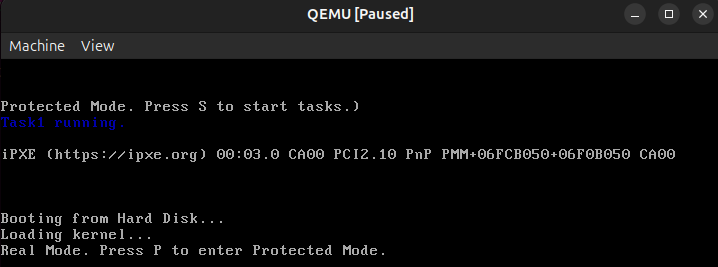


Рисунок 3.1 – Запуск первой задачи

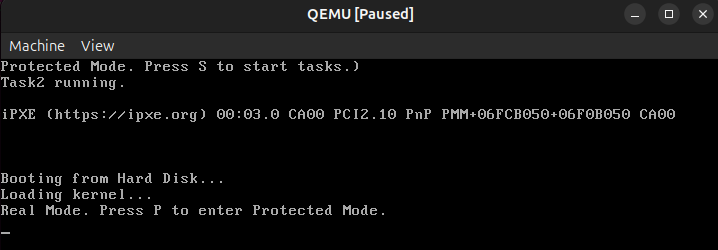


Рисунок 4.2 – Запуск второй задачи

# ВЫВОД

Реализация мультизадачности в защищённом режиме позволила на практике изучить принципы организации выполнения нескольких задач в 32-битной среде. Программа успешно демонстрирует ручное переключение между задачами, каждая из которых выполняет вывод информации с задержкой, имитируя параллельную работу. Это подтверждает корректную настройку сегментных регистров, видеопамяти и управление потоком исполнения. Полученные результаты показывают, что защищённый режим предоставляет широкие возможности для управления памятью, изоляции задач и реализации многозадачной архитектуры, что является основой современных операционных систем.