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 лабораторной работе № 2

на тему «Мультизадачность в защищённом режиме»

Выполнил:

студент гр. 153503

Киселёва Е.А.

Проверил:

Калиновская А.А.

Минск 2023

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc146631498)

[2 Краткие теоретические сведения 4](#_Toc146631499)

[3 Результаты выполнения лабораторной работы 5](#_Toc146631500)

[Выводы 6](#_Toc146631501)

[Список использованных источников 7](#_Toc146631502)

[Приложение А (обязательное) Листинг кода 8](#_Toc146631503)

## 1 ПОСТАНОВКА ЗАДАЧИ

Целью выполнения данной лабораторной работы является изучение принципов и средств реализации мультизадачности в защищенном режиме процессора. Написать программу, реализующую мультизадачность в защищенном режиме. Программа должна переключить процессор в защищенный режим, а затем запустить на выполнение 2-3 задачи, которые должны выполняться параллельно. Каждая задача выводит на экран свое сообщение. Задача выводит на экран часть сообщения, затем происходит переключение на другую задачу и т.д. Когда все задачи отработают программа должна вернуть процессор в реальный режим процессора и написание программы, переключающую процессор в защищенный режим, выводящую на экране сообщение и затем возвращающую процессор в реальный режим.

## 2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Под мультизадачностью подразумевают способность компьютера выполнять несколько задач одновременно. На самом деле процессор некоторое время выполняет один командный поток, затем быстро переключается на второй и выполняет его, переключается на третий и т.д. При этом при каждом переключении сохраняется контекст прерываемого потока, так что потом процессор сможет "безболезненно" продолжить выполнение прерванного потока команд. Благодаря высокому быстродействию создается иллюзия того, что все задачи выполняются одновременно (параллельно).

Для управления мультизадачностью нет специальных команд. Задачи переключаются командами FAR CALL, FAR JMP, INT, IRET. Однако при этом участвуют специальные дескрипторы: дескриптор сегмента состояния задачи (Task State Segment) и дескриптор шлюза задачи. Когда управление передается на один из таких дескрипторов, происходит переключение задачи. При переключении задачи процессор сохраняет (восстанавливает) свой контекст в сегменте состояния задачи (TSS). Селектор TSS выполняемой задачи хранится в регистре задачи (Task Register). При переключении задачи процессор может сменить LDT, что позволяет назначить каждой задаче свое адресное пространство, недоступное для других задач. Можно также перегрузить CR3, что позволяет применить для изолирования задач механизм страничного преобразования.

Защищённый режим. Как утверждает Intel, это "родной" (native) режим 32-разрядного процессора. В защищённый режим процессор надо переводить специальными операциями над системными регистрами и войти в этот режим процессор может только из реального режима. При работе в защищённом режиме процессор контролирует практически все действия программ и позволяет разделить операционную систему, драйвера и прикладные программы разными уровнями привилегий. Благодаря этому ОС может ограничить области памяти, выделяемой программам, запретить или разрешить ввод/вывод по любым адресам, управлять прерываниями и многое другое. При попытке программы выйти за допустимый диапазон адресов памяти, выделенной ей, либо при обращении к "запрещённым" для неё портам процессор будет генерировать исключения – специальный тип прерываний. Грамотно оперируя исключениями, операционная система может контролировать действия программ, организовать систему виртуальной памяти, мультизадачность и другие программные технологии.

## 3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ     РАБОТЫ

В рамках выполнения лабораторной работы было создано приложение, переключающее процессор в защищенный режим, затем запускающее на выполнение 2 задачи, которые выполняются параллельно. Каждая задача выводит на экран свое сообщение (Рисунок 1).

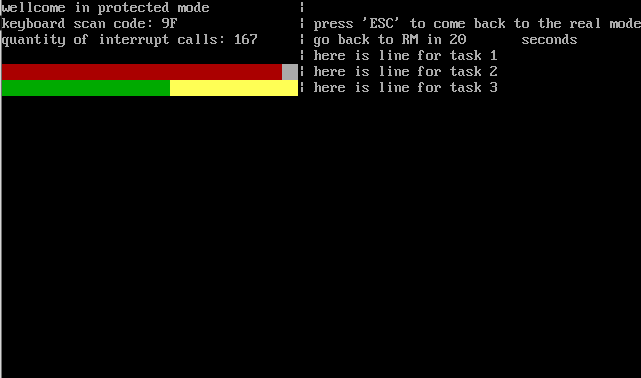


Рисунок 1 – Результат работы программы

## ВЫВОДЫ

В результате выполнения данной лабораторной работы были изучены принципы и средства реализации мультизадачности в защищенном режиме процессора. Было создано приложение, переключающее процессор в защищенный режим, затем запускающее на выполнение 2 задачи, которые выполняются параллельно.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мультизадачность [Электронный ресурс]. – Режим доступа: <http://www.sasm.narod.ru/docs/pm/pm_tss/chap_4.htm>.
2. Мультизадачность в процессоре I80386 [Электронный ресурс]. – Режим доступа: https://frolov-lib.ru/books/bsp.old/v06/ch4.htm.

**ПРИЛОЖЕНИЕ А**

## (обязательное)

## Листинг кода

Листинг 1 **–** Файлtasks.asm:

CODE\_2 segment para use32 ;Сегмент кода второй задачи - выводит красно-белую полосу слева-направо

assume cs:CODE\_2

CS\_2\_BEGIN = $

TASK\_2 proc

mov ax,DATA\_DESC

mov ds,ax

mov ax,TEXT\_DESC

mov es,ax

TASK\_2\_START:

mov edi,640

mov ecx,37

TASK\_2\_LOOP\_FILL: ;Цикл по выводу красной полосы

mov al,0DBh

mov es:[di],al

inc di

mov al,74h

mov es:[di],al

inc di

mov bx,cx

mov cx,60000

TASK\_2\_SLEEP\_0:

rept 90

nop

endm

loop TASK\_2\_SLEEP\_0

mov cx,bx

loop TASK\_2\_LOOP\_FILL

mov edi,640

mov ecx,37

TASK\_2\_LOOP\_CLEAN: ;Цикл по выводу белой полосы

mov al,' '

mov es:[di],al

inc di

mov al,74h

mov es:[di],al

inc di

mov bx,cx

mov cx,60000

TASK\_2\_SLEEP\_1:

rept 90

nop

endm

loop TASK\_2\_SLEEP\_1

mov cx,bx

loop TASK\_2\_LOOP\_CLEAN

TASK\_2\_END:

jmp TASK\_2\_START

TASK\_2 endp

SIZE\_CS\_2 = ($ - CS\_2\_BEGIN)

CODE\_2 ends

CODE\_3 segment para use32 ;Сегмент кода 3 задачи - выводит зелено-желтую полосу справа-налево

assume cs:CODE\_3

CS\_3\_BEGIN = $

TASK\_3 proc near

mov ax,DATA\_DESC

mov ds,ax

mov ax,TEXT\_DESC

mov es,ax

TASK\_3\_START:

mov edi,872

mov ecx,37

TASK\_3\_LOOP\_FILL: ;Цикл по выводу зеленой полосы

mov al,0DBh

mov es:[di],al

inc di

mov al,46

mov es:[di],al

sub di,3

mov bx,cx

mov cx,60000

TASK\_3\_SLEEP\_0:

rept 90

nop

endm

loop TASK\_3\_SLEEP\_0

mov cx,bx

loop TASK\_3\_LOOP\_FILL

mov edi,872

mov ecx,37

TASK\_3\_LOOP\_CLEAN: ;Цикл по выводу желтой полосы

mov al,' '

mov es:[di],al

inc di

mov al,46

mov es:[di],al

sub di,3

mov bx,cx

mov cx,60000

TASK\_3\_SLEEP\_1:

rept 90

nop

endm

loop TASK\_3\_SLEEP\_1

mov cx,bx

loop TASK\_3\_LOOP\_CLEAN

TASK\_3\_END:

jmp TASK\_3\_START

TASK\_3 endp

SIZE\_CS\_3 = ($ - CS\_3\_BEGIN)

CODE\_3 ends

STCK\_2 segment para

db 1000h DUP (?)

STCK\_2 ends

STCK\_3 segment para

db 1000h DUP (?)

STCK\_3 ends