

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»
Кафедра информатики

Отчет по лабораторной работе №2
Мультизадачность в защищенном режиме

Выполнил:
студент группы 053505

Слущкий Н. С.

Проверила:
Калиновская А. А.

Минск 2022

СОДЕРЖАНИЕ

1. Цель работы
2. Контрольные вопросы
3. Программная реализация
4. Выводы

1. Цель работы

Изучить принципы и средства реализации мультизадачности в защищенном режиме процессора. Получить практические навыки по программированию и использованию этих средств.

2. Постановка задачи

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.

3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какая обязательная информация сохраняется в сегменте состояния задачи?

Сегмент состояния задачи состоит из двух частей. Обязательная часть TSS объемом 104 байта содержит информацию, необходимую для рестарта данной задачи после ее вызова на исполнение, а также некоторую другую информацию.

2. Какая дополнительная информация может быть сохранена в сегменте состояния задачи?

Дополнительная часть может содержать какую-либо информацию о задаче, используемую операционной системой (имя задачи, комментарии и т. д.)

3. Для чего используется битовая карта ввода/вывода?

Битовая карта ввода/вывода процессора i80386 позволяет для каждой задачи определить порты, которые эта задача может использовать. То есть операционная система имеет возможность санкционировать любую задачу для использования любого набора адресов портов ввода/вывода. Если задача попытается обратиться к несанкционированному порту ввода/вывода, произойдёт исключение.

4. Какие существуют в защищенном режиме способы переключения задач? Сравните их.

Помимо простого переключения задач, процессор 80386/80286 предлагает еще два средства организации мультизадачности:

1. Переключение задачи может выполняться вследствие прерываний и исключений. Процессор не только выполняет переключение задач для обработки прерывания или исключения, но и автоматическое переключение назад, на прерванную задачу, после возврата из прерывания или исключения. Прерывания могут происходить и во время задач обработки прерывания.

2. При каждом переключении на другую задачу процессор 80386/80286 загружает новое значение в регистр LDTR, иными словами, каждая задача имеет собственную локальную дескрипторную таблицу. Это может использоваться для того, чтобы дать каждой задаче собственное отображение логических адресов в физические. Тем самым обеспечивается дополнительное средство защиты, поскольку задачи могут быть таким образом изолированы, и их взаимное влияние друг на друга исключено.

5. Какие действия выполняет процессор при переключении задач?

1. Для команд CALL и JMP проверяет привилегии (CPL текущей задачи и RPL селектора новой задачи не могут быть больше, чем DPL шлюза или TSS, на который передается управление).

2. Проверяется дескриптор TSS (его бит присутствия и лимит).

3. Проверяется, что новый TSS, старый TSS и все дескрипторы сегментов находятся в страницах, отмеченных как присутствующие.

4. Сохраняется состояние задачи.

5. Загружается регистр TR. Если на следующих шагах происходит исключение, его обработчику придется доделывать переключение задач, вместо того чтобы повторять ошибочную команду.

6. Тип новой задачи в дескрипторе изменяется на занятый, и флаг TS устанавливается в CRO.

7. Загружается состояние задачи из нового TSS: LDTR, CR3, EFLAGS, EIP, регистры общего назначения и сегментные регистры.

3. Программная реализация

3.1 При запуске итоговая рабочая программа начинает выводить несколько текстов. Тесты выводятся не одновременно, а поочередно, останавливая выполнение предыдущей задачи перед выполнением своей.

Для компиляции использован компилятор TASM и TLINK. Чтобы собрать программу без ошибок, понадобилось передавать компилятору tasm флаг /m, а линкеру tlink флаг /3. Всё это происходило в эмуляторе – программе DOSBox.

5. Выводы

В ходе выполнения лабораторной работы были усвоены основы реализации мультизадачной работы процессора, практике во встроенном эмуляторе в VS Code был написан код, а затем вручную скомпилирован и слинкован в среде DOSBox. Код вначале переключает в защищенный режим, а далее параллельно выполняются 3 задачи, которые выводят различные тексты. Тексты выводятся параллельно, путем остановки одной из задач и передачи управления следующей и так до тех пор, пока не будут выведены все тексты.

На основании полученных результатов цели лабораторной работы можно считать достигнутыми.



ommand: masm runner start

escription: Starts MASM CMD to allow compilation and running of code

Starting Webview

