

Факультет: «Автоматики и электроники».

Кафедра №27.

ОТЧЕТ
к лабораторной работе
по курсу «Микропроцессоры архитектуры MIPS»
«ОСРВ 2000»

Выполнили студенты
Смирнов Владимир Юрьевич (А9-11)
Полинкин Алексей Павлович (А9-11)
Соломатин Анатолий Владленович (А9-09)
Облов Константин Юрьевич (А9-11)

Преподаватель: Тихонов Юрий Николаевич

Москва, 2010 г.

Содержание

1. Цель работы	3
2. Введение	3
3. Порядок выполнения лабораторной работы	3
А. Текст программы на языке Си.	4
В. Конфигурирование ОСРВ	4

1. Цель работы

Разработка специального программного обеспечения для операционной системы реального времени.

2. Введение

Операционная система ОС-2000 предназначена для создания программного обеспечения вычислительных систем, комплексов и средств автоматизированного управления, работающих в режиме реального масштаба времени. Большая часть интерфейсов стандарта POSIX (the Portable Operating System Interface/Интерфейсы Мобильной Операционной Системы) 1003.b, определяющего сервисы операционных систем реального времени, реализована в ОС2000. Все функции и интерфейсы, реализованные в ОС-2000, POSIX-совместимы.

3. Порядок выполнения лабораторной работы

- Написана программа на языке Си. А
- Сконфигурировано ядро так, чтобы оно включало исходный код нашей программы. В
- Скомпилировано ядро с созданием бинарного файла определённого формата.
- Подключения стэнда к компьютеру посредством двух СОМ-портов и включение его.
- Изменяем скорость работы СОМ-порта на максимальное значение 115200 с помощью команды *sbaud a 115200* в *prime* и *set baud 115200* в *kermit*. Это необходимо для увеличения скорости загрузки прошивки.
- Подготавливаем стэнд к получению нового ядра: *lz 0x80100000 a*. А в *kermit* устанавливаем протокол *zmodem*: *set protocol zmodem* и затем запускаем передачу файла: *send os - 11.bin*. Особенности программы *Prime* состоят в том, что ждет она не очень продолжительное время, поэтому, в случае возникновения события *TIMEOUT*, необходимо заново подсоединиться к *Prime* командой *connect* и заново перевести его в режим приема данных командой *lz 0x80100000 a*.
- После передачи файла необходимо записать его во флэш. Для этого необходимо запомнить размер принятого файла (в нашем случае он составлял 0x16CABC) и затем выполнить *wfn 0x80100000 < размер файла > 0xb8300000*. Подтверждаем запись данных во внутреннюю flash-память.

- Устанавливаем скорость передачи в 9600 на ПК (в kermit), подключаемся к плате и перезагружаем стэнд.
- Запускаем выполнение ядра: `g 0xb8300000`
- Исполняем написанную нами функцию просто введя её имя.

Стоит отметить, что наш лабораторный стэнд постоянно показывал ошибки *RPC timeout for se* но на выполнение работы это не повлияло.

A. Текст программы на языке Си.

```
#include <stdio.h>

int hw()
{
printf("Simple test program: welcome to HELL!\n");
return 0;
}
```

B. Конфигурирование ОСРВ

1. Выполнить `xconfig.sh` (альтернативно `gmake xconfig`), чтобы открыть конфигуратор
2. После подключения включения нашей программы (`app.c`) нужно сохранить изменения конфигурации и выполнить `gmake`
3. В файле `mb_bin.sh` поменять `RESULT` на `os-11.bin`
4. Выполнить `mb_bin.sh`. На выходе получится файл `os-11.bin`, содержащий нашу программу.