SE 350 Project
Real Time Executive (RTX)

Based on M. Akon's presentation in Spring'08

Project outline

- RTX – Real Time Executive
  - Microkernel based operating system
  - MCF5307 based micro-controller board
- Challenges
  - Handle H/W directly – (Mixed Programming)--assembly
  - Large code base
  - Difficult to debug
  - Recall ECE-250 / SE-240 & ECE-222

Features

- Basic multiprogramming environment
- 5 priority levels---- 4 user priority levels plus one for Null process
- Preemption
- Simple memory management
- Interprocess communication (message)
- Basic timing service
- Console I/O
- Debug support – hot key
- System processes: command decoder, display
- User processes – wall clock display and others

• i-processes to handle interrupts

Features ...

- A co-operative, non-malicious environment
- User and supervisor modes
- 1 Megabyte of memory

Out of scope

- Paging / segmentation
- Virtual memory
- Dynamic loading of processes
- Termination of processes
- File system
- Time slicing

Deliverables

- Part 1
- A software design document (SDD)
  - Outlining the design for your RTX
- Demo
  - Demonstrate various features from the requirements
  - Test processes – provided during demo
- Final report
  - Describing changes to your design
  - Lessons learned

➔ Both soft- and hard-copy submission (except code)

Software Design Document

- Requirements
  - A structural description of the design
  - Functional description of all procedures
  - Implementation, testing, and measurement plan
- Why important?
  - Problems during implementation is inversely proportional to efforts in preparing this document

Format

- Main body - not more than 30 pages
- Use standard formatting
  - Title page
  - Table of content
  - List of figures and tables
  - Organized into sections and subsection
- Use figures and tables

Document Outline

- Introduction
- Global Information
- Primitives
- Software Interrupts
- Hardware Interrupts
- System Processes
- User Processes
- Initialization
- Implementation / Test Plan

Introduction

- Basic information and overview: we are building an OS, this is the design doc, etc.
- Overview of your operating system's structure (largely determined by the project description)

Global information

- List and define the meaning of:
  - Data structures (eg. how you're storing process queues, lists of memory blocks, etc.)
  - Constants (process IDs, states, etc.)
  - Global variables
- Memory map: A diagram showing the 5307's memory, and where in memory various elements of your operating system will reside.
  - Remember the 1 MB limitation

Primitives

- Basic services provided by the microkernel
- Pseudocode
  - No C code!
  - May use the style from ECE-250 / SE-240
  - Should be detailed enough to allow TAs to understand your intended implementation

H/W, S/W interrupts

- Areas where switch between user and supervisor mode
- Software interrupt section essentially describes the interface by which processes invoke the kernel services
- Describe the path of execution from the interrupt vector to each i-process
- Describe i-processes – pseudocode, no H/W details

System & user processes

- Describe the purpose of each process, assumptions, requirements, dependencies on other processes
- Describe the format of the messages the process can receive / send
- Pseudocode for each process

Initialization

- Outline the steps the OS will take when it starts execution
- You are required to make certain system parameters tunable: how are these initialized?

Implementation / test plan

- WHO will complete WHAT code and WHEN?
- WHEN will WHAT code be integrated?
- WHEN will various functionality be working?
- HOW will you be testing your code?
  - Manual test cases? An automated test suite?
  - Describe your test cases!
- WHERE and HOW will your code be stored?
  - Source control? Stored on a central computer?
- HOW will your code be compiled?
  - Dependency graph of your build system? – Makefile
- Development tools

Advice

- Start early
- Put serious effort in preparing the software design document
- Emphasize on your assumption in the design document
- Form your group carefully
  - Be honest with each other about your various strengths and weaknesses
- You may face long debugging sessions
  - Make lots of debug information available
  - #ifdef

Advice ...

- Start with
  - Project design description
  - SBC5307 Manual & Reference
  - Understand, compile and run the code from the manual
- Consult the manual before asking question about the micro-controller
- Get help from TA's early on