Introduction to Operating Systems
Project #0: Overview

04/07/2022

Prof. Seongsoo Hong
sshong@redwood.snu.ac.kr
SNU RTOSLab
Dept. of Electrical and Computer Engineering
Seoul National University

Agenda

I. **Overview of Project**
II. Introduction to eOS
III. Development Environment
IV. Project Schedule

프로젝트의 목표와 내용

목표

- 서울대학교 실시간 운영체제 연구실에서 개발한 eOS를 사용하여 운영체제의 다양한 기능들을 구현한다
- 이 과정을 통해 운영체제의 내부 구조와 핵심적인 기능들이 어떻게 구현되어 있는지 학습한다

내용

- 배포될 eOS는 함수들의 prototype만 제공된다
- 이 함수들의 내용을 채워서 운영체제를 완성한다

Agenda

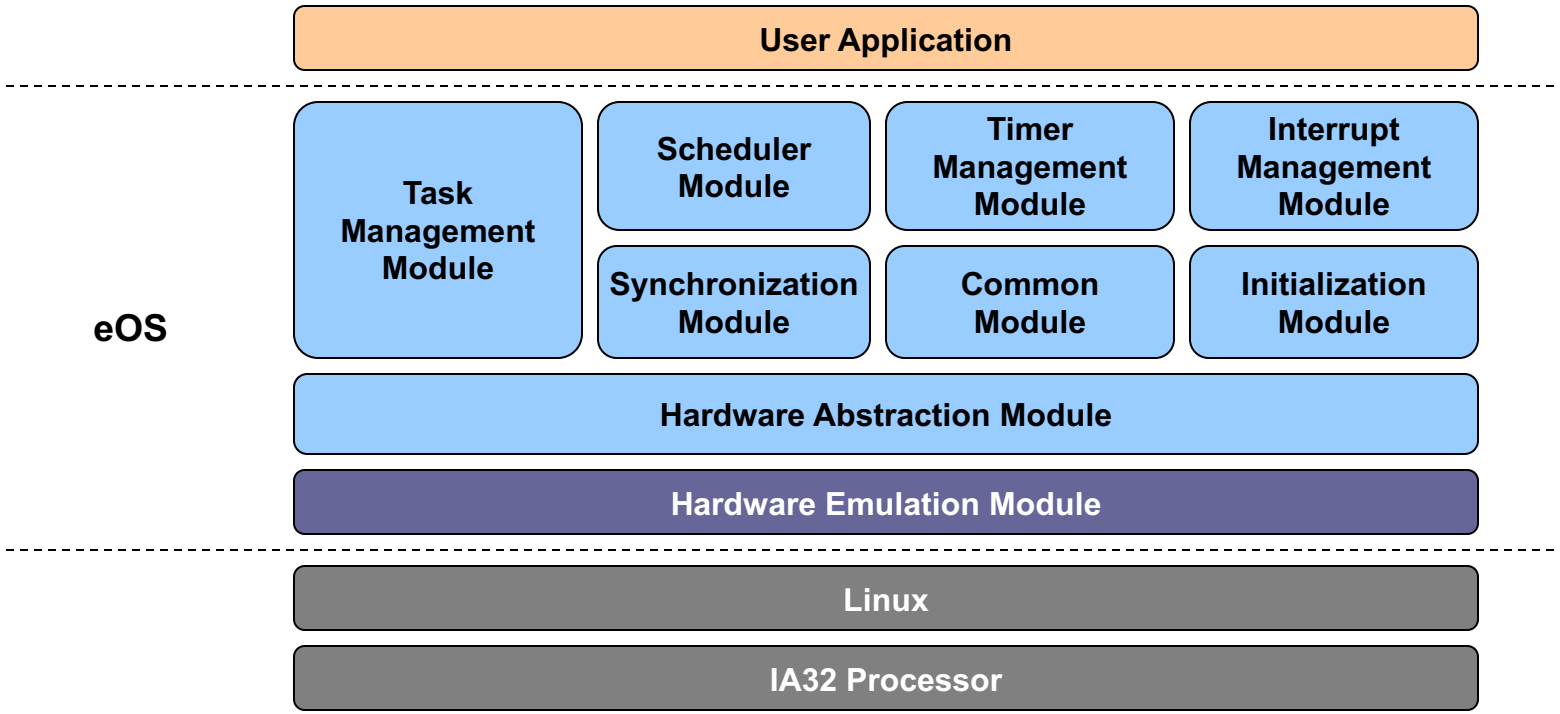
I. Overview of Project
II. **Introduction to eOS**
III. Development Environment
IV. Project Schedule

What is eOS? (1)

- Educational Operating System (eOS)
  - Real-time, preemptive, multitasking kernel developed for RTOS education
  - Extremely portable
    - Most of the kernel code is written in ANSI C
    - HAL만 교체하면 다른 HW 상에서도 사용 가능
  - Developed at RTOS Lab. of SNU
    - http://redwood.snu.ac.kr

What is eOS? (2)

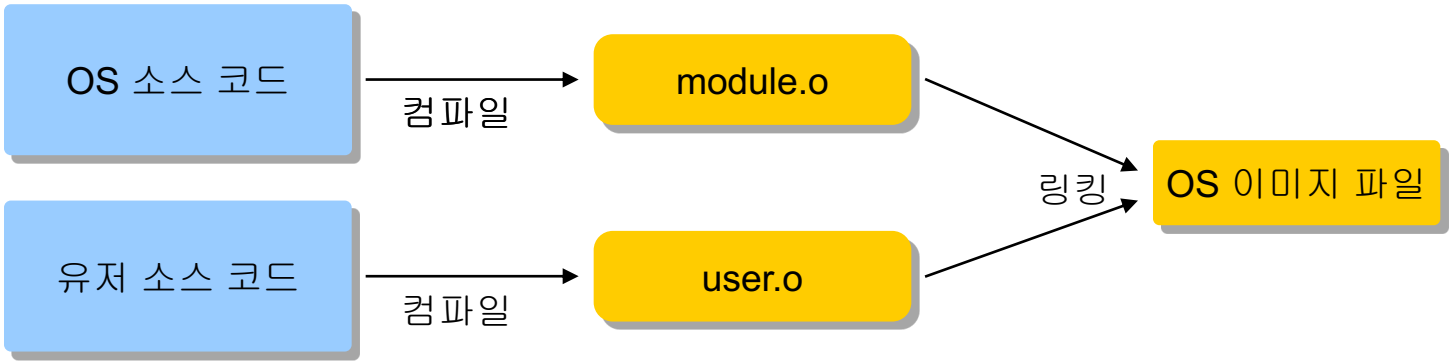
- Layered architecture of eOS on Linux



What is eOS? (3)

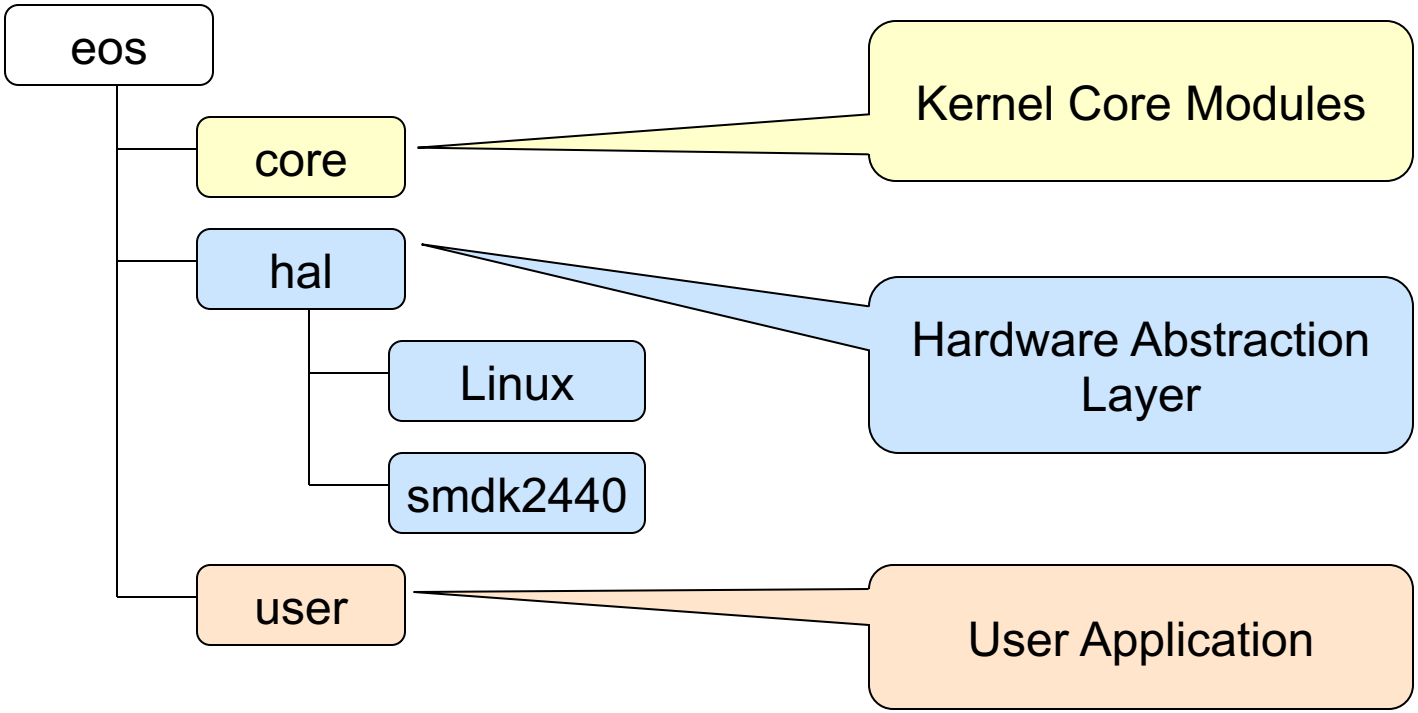
- Execution environment
  - eOS는 Linux 위에서 하나의 프로세스로 실행된다

- eOS is a library kernel
  - 유저 코드와 커널 코드가 링크되어 하나의 executable file이 된다
  - 커널 코드는 라이브러리 형태로 제공된다



What is eOS? (4)

- Source tree



What is eOS? (5)

- Kernel core modules: eos/core/

Header files	
`eos.h`	Defining all external APIs and data types
`eos_internal.h`	Defining all functions and data types for internal use
C files	
`common.c`	Common functions such as serial output and list manipulation
`interrupt.c`	Interrupt management module
`main.c`	Kernel initialization module
`scheduler.c`	O(1) scheduler module
`sync.c`	Synchronization module
`comm.c`	Communication module
`task.c`	Task management module
`timer.c`	Timer management module

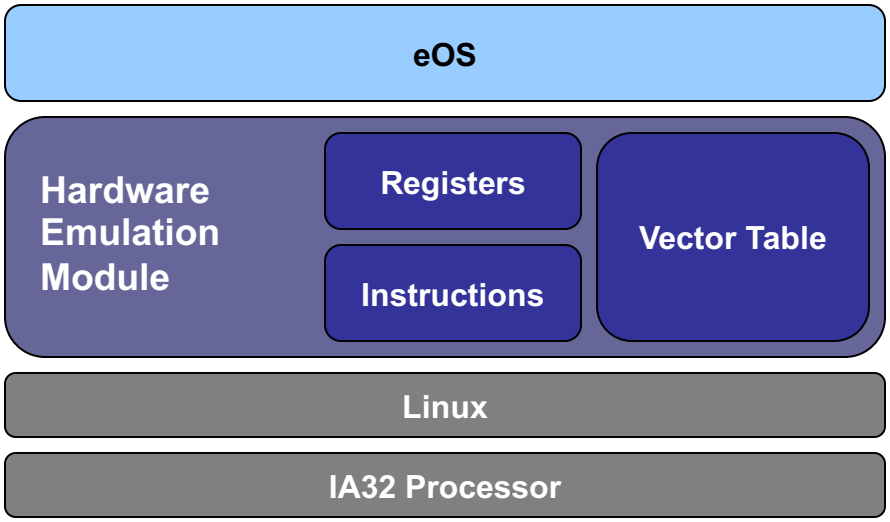
What is eOS? (6)

- HAL for Linux: eos/hal/linux

Header files	
`emulator.h`	Defining vector table and registers for hardware emulation
`emulator_asm.h`	Defining instructions for hardware emulation
`include.h`	Defining architecture specific paremeters
`type.h`	Data type definitions
C files	
`context.c`	Implementing context switching
`init.c`	Initializing HAL
`interrupt.c`	Acknowledging and masking IRQs
`serial.c`	Implementing serial output
Assembly files	
`entry.S`	Calling vector table entries
`interrupt_asm.S`	Enabling and Disabling of interrupts

What is eOS? (7)

- Hardware emulation module
  - Used for IA32/Linux HAL
  - Emulating hardware so that eOS can operate on top of both Linux and IA32 processor



What is eOS? (8)

- Hardware emulation module: “vector table”
  - When hardware event occurs, jumps to the address saved in the corresponding vector table entry
  - Currently 4 entries exist
    - `_vector[0]`: system reset (initialization)
    - `_vector[1]`: not used
    - `_vector[2]`: not used
    - `_vector[3]`: interrupt request

What is eOS? (9)

- Hardware emulation module: “registers”
  - `_eflags`: Status register (interrupt enable/disable)
  - `_irq_pending`: Interrupt pending register
  - `_irq_mask`: Interrupt mask register
- Hardware emulation module: “instructions”
  - `_CLI`: Disabling interrupts
  - `_STI`: Enabling interrupts
  - `_IRET`: Popping instruction pointer from stack

What is eOS? (10)

- Naming rules for functions and variables
  - 커널 함수와 전역 변수의 이름은 “eos_” 혹은 “_os_” 접두사로 시작한다
    - 유저가 사용할 수 있는 함수나 변수는 “eos_” 접두사로
    - 커널이 내부적으로 사용하는 함수나 변수는 “_os_” 접두사로
  - 단어와 단어 사이는 “_”로 구별한다
  - 어떠한 역할을 하는지 알 수 있도록 영문법의 어순을 따른다
    - ‘동사 + 목적어’ 혹은
    - ‘형용사 + 명사’

What is eOS? (11)

- Data types

Data Type Name	
bool_t	typedef unsigned char
int8u_t	typedef unsigned char
int8s_t	typedef signed char
int16u_t	typedef unsigned short
int16s_t	typedef signed short
int32u_t	typedef unsigned int
int32s_t	typedef signed int
fp32_t	typedef float
fp64_t	typedef double
addr_t	typedef void*
size_t	typedef unsigned int

Agenda

I. Overview of Project
II. Introduction to eOS
III. **Development Environment**
IV. Project Schedule

Build/Execution Environment

- Ubuntu Linux

- 설치 절차
  - VMware Player(또는 Virtual Box) 설치
  - Ubuntu 16.04.6 LTS 32bit iso 파일 다운로드 (http://releases.ubuntu.com/xenial/ubuntu-16.04.6-desktop-i386.iso)
  - VMware에서 새로운 virtual machine 만들기 선택 후 Ubuntu 설치(2에서 다운받은 iso 파일 선택)
  - VMware에서 공유폴더 설정하거나, 네트워크 활용하여 eos.tar.gz 파일 전송 받아 프로젝트 수행

Kernel Build

- Build sequence
  - Download the kernel
  - Unzip the compressed archive file “eos.tar.gz”
    - `tar –xvf eos.tar.gz`
  - Move to the eOS home directory and build
    - `cd eos`
    - `make clean`
    - `make all`

Agenda

I. Overview of Project
II. Introduction to eOS
III. Development Environment
IV. **Project Schedule**

Subprojects (1)

- Project 1
  - eOS 코드 분석
  - I386 architecture and calling convention 분석
- Project 2
  - Multi-tasking

Subprojects (2)

- Project 3
  - Periodic task
  - Priority scheduler
- Project 4
  - Synchronization primitives
  - Communication primitives

Grading Policy

- 팀 구성
  - 1인 1팀
- 채점
  - 딜레이: 1일에 20%씩 점수 차감
  - 카피 적발 시 0점 처리

Question or Comment?