PGT200 - Operating Systems

This course is *Operating Systems*, offered by the Department of Electronics Engineering Technology.

Announcement

[20161007] Just updated 😛 Some info are still from the previous offering…

Lecture Slides

- Lecture 0 - Course Introduction

These are from Tanenbaum's Book:

- Chapter 1 - Introduction
- Chapter 2 - Processes and Threads
- Chapter 3 - Memory management
- Chapter 4 - File systems
- Chapter 5 - Input/Output

Lab Notes

- Lab Work 1 - Tools and Platform
- Lab Work 2 - Processes and Threads (Part 1,Part 2)
- Lab Work 3 - Bootstrap Loader
- Lab Work 4 - Simple Kernel

Useful Links

- OSDev Wiki
- OSDever Tutorials
- Custom Bootloader - OSDev Wiki
- Higher Half Kernel - OSDev Wiki

Assignments

- Assignment 1:

  pgt200_201617s1_assign1.txt

```
------------
ASSIGNMENT 1
------------

The Dining Philosopher's Problem

You are to implement a solution for the dining philosophers
problem using any
method you see fit. The number of philosophers is fixed at 5 and
each must run
on its own separate process. Based on your proposed solution,

a) explain your implementation of the chosen algorithm. [10 marks]

b) argue which algorithm would be the best fit for this problem.
[10 marks]

c) answer general questions [5 marks]

This is an individual assignment. An example framework has been
prepared for
you to refer to.

ASSIGNMENT DUE: W07/14 17-21/10/2016 (Anytime during the week)
```

You can use this as a reference.

- Assignment 2:

pgt200_201617s1_assign2.txt

```
------------
ASSIGNMENT 2
------------

Evaluate any two existing filesystems in desktop OS (e.g. Linux,
BSD, Unix,
Windows, MacOS). Based on the given sub-topics, identify and
evaluate their
respective implementations.

a) General filesystem layout
b) File creation/deletion
c) Other features

This is an individual assignment.

ASSIGNMENT DUE: W12/14 {28/11-2/12}/2016 (Anytime during the week)
```

Lab Project

- The following project titles uses my1load86 and my1playos projects as a reference.

pgt200_201617s1_projects.txt

```
-------------------
LAB PROJECT TITLE 1
-------------------

Based on my1load86 project, write an x86 boot code (to replace
loader.asm)
that is capable of loading a second stage boot code LDPLAYOS.SYS
from a file
system other than FAT. Linux EXT file system is recommended but
others can be
proposed as well.

-------------------
LAB PROJECT TITLE 2
-------------------

Based on my1playos project, implement any OS tasks (e.g. process
management,
memory management,filesystem) that have been covered in this
course.

-----------------------
ASSESSMENT REQUIREMENTS
-----------------------

This is a group assignment - but, marks will be evaluated
individually.
Evaluation will be based on a demonstration and a simple report
(not MORE
than 10 pages) explaining your work.

PROJECT DUE: W14/14 12-16/12/2016 (Anytime of the week)
```

Course Synopsis

This course presents the fundamental concepts and design of operating systems. Students will be guided to explore basic operating system concepts like processes, memory management, scheduling, file systems and IO management. Basic design and implementation of some of these concepts will be covered during laboratory sessions.

Course Outcome

1. Able to discuss basic operating systems concepts
2. Able to implement basic operating systems concepts
3. Able to evaluate implementations of basic operating systems concepts

Course Assessment

	Examinations		**Course Work**		
Total Contribution	60%		40%		
Assessment	Mid-Term Examinations	Final Examinations	Assignments	Lab Assessment	Lab Project
Contribution	20%	40%	10%	20%	10%

Course Syllabus

Week	**Lecture**	**Laboratory**	**Notes**
Week 01	• Introduction to Operating Systems ◦ fundamental concepts ◦ implementations	Lab Work 1	
Week 02	• Processes & threads ◦ processes - concept and implementation ◦ threads - concept and implementation ◦ execution and scheduling	Lab Work 1 (cont.)	
Week 03	• Processes & threads (cont.) ◦ critical region, mutual exclusion ◦ implementing critical region ◦ solutions for mutual exclusion	Lab Work 2	
Week 04	• Memory Management ◦ address space & memory abstraction ◦ virtual memory (paging)	Lab Work 2 (cont.)	
Week 05	• Memory Management (cont.) ◦ page replacement algorithm ◦ segmentation (implementation and issues)	Lab Work 2 (cont.)	**Assignment 1 Queue**

Week	Lecture	Laboratory	Notes
Week 06	• Bootstrapping ∘ bootstrapping concepts ∘ x86 PC boot sequence ∘ x86 PC programming	Lab Work 3	**Lab Assessment 1 (10%)**
Week 07	• Bootstrapping (cont.) ∘ x86 PC programming (cont.)	Lab Work 3 (cont.)	**Assignment 1 Due (5%)**
Week 08	• Bootstrapping (cont.) ∘ x86 PC programming (cont.)	Lab Work 3 (cont.)	**Mid-term Examination (20%)** **Assignment 2 Queue**
Week 09	• Filesystem Management ∘ files and directories ∘ implementation	Lab Work 4	
Week 10	• Filesystem Management (cont.) ∘ management and optimization	Lab Work 4 (cont.)	**Assignment 2 Due (5%)**
Week 11	• I/O Management • devices and controllers • device drivers	Lab Project	**Lab Assessment 2 (10%)**
Week 12	• I/O Management (cont.) ∘ DMA, memory-mapped I/O ∘ programmed and interrupt-driven I/O	Lab Project (cont.)	
Week 13	• Operating System Design ∘ interface design ∘ implementation ∘ management trends	Lab Project (cont.)	
Week 14	• Modern Operating Systems ∘ general overview of currently used OS ■ Unix, Linux, BSD, Windows, MacOSX	Lab Project (cont.)	**Lab Project (10%)**