



Operating Systems (CMPS 272)

Lecture 1: Overview

Mohamad Jaber

American University of Beirut

Faculty of Arts & Sciences - Department of Computer Science

August 30, 2017

Outline

- 1 Administrative Details
- 2 Course Description, Objectives, and Contents
- 3 Contents
- 4 Examples of Operating Systems

- 1 Administrative Details
- 2 Course Description, Objectives, and Contents
- 3 Contents
- 4 Examples of Operating Systems

Useful Info.

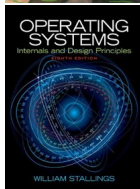
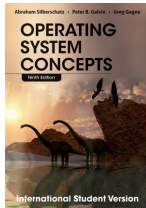
Instructor	Mohamad Jaber
Office	Bliss Hall, 312A
Phone	4256
Email	mj54@aub.edu.lb
Office Hours	MW 9:00-10:30AM, or by appointment

Teaching Assistant	Julia El Zini
Office	Bliss Graphics Lab
Email	jwe04@mail.aub.edu
Office Hours	MW 3:00-4:15PM or by appointment

Resources

The course will use the following two **textbooks** as references:

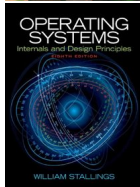
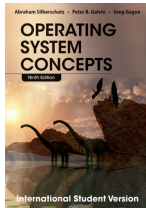
- Operating System Concepts, 9/e, Silberschatz, Galvin, and Gagne, Wiley (<http://os-book.com>) - assigned for the course
- Operating Systems: Internals & Design Principles, 8/e, William Stallings, Prentice Hall, (<http://williamstallings.com/OperatingSystems/>)



Resources

The course will use the following two **textbooks** as references:

- Operating System Concepts, 9/e, Silberschatz, Galvin, and Gagne, Wiley (<http://os-book.com>) - assigned for the course
- Operating Systems: Internals & Design Principles, 8/e, William Stallings, Prentice Hall, (<http://williamstallings.com/OperatingSystems/>)



Remark The course also uses some **online materials** and publications

Prerequisites

Prerequisites

The course assumes that you are comfortable with programming in C under Linux

Prerequisites

Prerequisites

The course assumes that you are comfortable with programming in C under Linux

Remark If not, please refer to Moodle or online materials - **Do not hesitate to ask me!**

Evaluation

Participation and Attendance	5%
Assignments and Projects	30%
Midterm (October 23 - Tentative)	30%
Final Exam (TBA)	35%
Total	100%

Evaluation

Participation and Attendance	5%
Assignments and Projects	30%
Midterm (October 23 - Tentative)	30%
Final Exam (TBA)	35%
Total	100%

- If you miss the midterm you should present a medical excuse and the final will count for 65%.

Evaluation

Participation and Attendance	5%
Assignments and Projects	30%
Midterm (October 23 - Tentative)	30%
Final Exam (TBA)	35%
Total	100%

- If you miss the midterm you should present a medical excuse and the final will count for 65%.
- The midterm and the final exam are **cumulative**, i.e., it will cover all class material to date.

Assignments/Projects

- Some of the assignments will **not be graded**; but I will ask questions about them during the exams or small quizzes.

Assignments/Projects

- Some of the assignments will **not be graded**; but I will ask questions about them during the exams or small quizzes.
- Although you are encouraged to discuss assignments, all work handed in must be done **independently**.
- Copying from each other will result in a **zero** grade for both copies.

Assignments/Projects

- Some of the assignments will **not be graded**; but I will ask questions about them during the exams or small quizzes.
- Although you are encouraged to discuss assignments, all work handed in must be done **independently**.
- Copying from each other will result in a **zero** grade for both copies.
- The minimum grade of assignments is **40**, except for plagiarism it is zero.

Assignments/Projects

- Some of the assignments will **not be graded**; but I will ask questions about them during the exams or small quizzes.
- Although you are encouraged to discuss assignments, all work handed in must be done **independently**.
- Copying from each other will result in a **zero** grade for both copies.
- The minimum grade of assignments is **40**, except for plagiarism it is zero.
- Your code must be **easy to read**:
 - Indent code, keep lines and (when possible) functions short.
 - Use a uniform coding style.
 - Put comments on structure members, globals, functions.

Attendance

- You are expected to **attend** lectures.

Be aware of AUB regulations regarding attendance!

Attendance

- You are expected to **attend** lectures.
- **Lectures are in sequence.**

Be aware of AUB regulations regarding attendance!

- 1 Administrative Details
- 2 Course Description, Objectives, and Contents**
- 3 Contents
- 4 Examples of Operating Systems

Course Description and Objectives

- To provide an introduction to the fundamentals of **operating system** function, design, and implementation.

Course Description and Objectives

- To provide an introduction to the fundamentals of **operating system** function, design, and implementation.
- The course divides into **three major** sections:

Course Description and Objectives

- To provide an introduction to the fundamentals of **operating system** function, design, and implementation.
- The course divides into **three major** sections:
 - Concurrency (processes, threads, scheduling, synchronization, and deadlocks).

Course Description and Objectives

- To provide an introduction to the fundamentals of **operating system** function, design, and implementation.
- The course divides into **three major** sections:
 - Concurrency (processes, threads, scheduling, synchronization, and deadlocks).
 - Memory management (memory management strategies and virtual memory management).

Course Description and Objectives

- To provide an introduction to the fundamentals of **operating system** function, design, and implementation.
- The course divides into **three major** sections:
 - Concurrency (processes, threads, scheduling, synchronization, and deadlocks).
 - Memory management (memory management strategies and virtual memory management).
 - File systems, Secondary Storage, I/O systems.

Course Description and Objectives

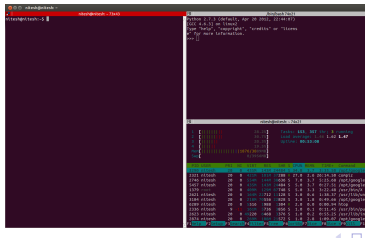
- To provide an introduction to the fundamentals of **operating system** function, design, and implementation.
- The course divides into **three major** sections:
 - Concurrency (processes, threads, scheduling, synchronization, and deadlocks).
 - Memory management (memory management strategies and virtual memory management).
 - File systems, Secondary Storage, I/O systems.

*Understanding the OS makes you a more **effective programmer!***

Learning Outcomes

Upon completion of this course, you will be able to:

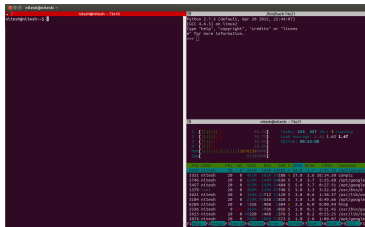
- 1 List the **objectives** and **functions** of modern Operating Systems.



Learning Outcomes

Upon completion of this course, you will be able to:

- 1 List the **objectives** and **functions** of modern Operating Systems.
- 2 Describe the concepts of **processes**, **threads**.

A screenshot of a Linux terminal window. The terminal shows the output of the command 'cat /etc/passwd'. The output displays system accounts (root, daemon, bin, etc.) and user accounts (systemd, dbus, etc.). The terminal is titled 'alim@alim:~\$' and the prompt is '\$ cat /etc/passwd'. The output is as follows:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/usr/sbin:/usr/sbin/nologin
systemd:x:4:4:systemd:/lib/daemon:/usr/sbin/nologin
dbus:x:8:8:dbus:/usr/bin:/usr/sbin/nologin
systemd+_:x:9:9:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:10:10:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:11:11:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:12:12:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:13:13:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:14:14:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:15:15:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:16:16:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:17:17:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:18:18:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:19:19:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:20:20:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:21:21:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:22:22:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:23:23:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:24:24:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:25:25:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:26:26:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:27:27:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:28:28:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:29:29:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:30:30:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:31:31:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:32:32:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:33:33:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:34:34:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:35:35:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:36:36:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:37:37:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:38:38:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:39:39:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:40:40:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:41:41:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:42:42:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:43:43:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:44:44:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:45:45:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:46:46:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:47:47:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:48:48:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:49:49:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:50:50:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:51:51:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:52:52:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:53:53:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:54:54:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:55:55:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:56:56:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:57:57:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:58:58:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:59:59:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:60:60:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:61:61:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:62:62:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:63:63:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:64:64:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:65:65:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:66:66:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:67:67:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:68:68:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:69:69:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:70:70:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:71:71:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:72:72:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:73:73:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:74:74:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:75:75:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:76:76:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:77:77:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:78:78:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:79:79:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:80:80:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:81:81:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:82:82:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:83:83:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:84:84:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:85:85:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:86:86:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:87:87:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:88:88:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:89:89:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:90:90:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:91:91:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:92:92:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:93:93:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:94:94:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:95:95:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:96:96:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:97:97:systemd+:/usr/bin:/usr/sbin/nologin
systemd+sd:x:98:98:systemd+sd:/usr/bin:/usr/sbin/nologin
systemd+_:x:99:99:systemd+:/usr/bin:/usr/sbin/nologin
```

Learning Outcomes

Upon completion of this course, you will be able to:

- 1 List the **objectives** and **functions** of modern Operating Systems.
- 2 Describe the concepts of **processes**, **threads**.
- 3 **Build** and **modify** Operating System **kernel** from source code.

The screenshot shows a terminal window with a dark background. The top part displays the output of the 'cat /etc/passwd' command, showing system users like 'root:x:0:0:root:/root:/bin/bash' and 'daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin'. Below this, there is a section for kernel parameters, including 'kernel.hostname = ubuntu', 'kernel.loglevel = 4', and 'kernel.logconsole = 0'. The bottom part of the terminal shows a list of processes with columns for PID, PPID, UID, GID, and command, such as '1 init 0 0 0 /usr/sbin/init' and '2 kthreadd 0 0 0 [kthreadd]'. The terminal title bar indicates the window is titled 'ubuntu - [Terminal]'. At the bottom right, there are standard terminal navigation icons.

Learning Outcomes

Upon completion of this course, you will be able to:

- 1 List the **objectives** and **functions** of modern Operating Systems.
- 2 Describe the concepts of **processes**, **threads**.
- 3 **Build** and **modify** Operating System **kernel** from source code.
- 4 Demonstrate **practical** experience under the Linux Operating System.

The screenshot shows a terminal window with the following content:

```

Linux 2.6.32-042.el5, Apr 20 2006, root@x86_64
root@x86_64 ~# cat /etc/passwd | grep root
root:x:0:0:root:/root:/bin/bash
root:x:1:1:root:/root:/bin/bash
root:x:2:2:root:/root:/bin/bash
root:x:3:3:root:/root:/bin/bash
root:x:4:4:root:/root:/bin/bash
root:x:5:5:root:/root:/bin/bash
root:x:6:6:root:/root:/bin/bash
root:x:7:7:root:/root:/bin/bash
root:x:8:8:root:/root:/bin/bash
root:x:9:9:root:/root:/bin/bash
root:x:10:10:root:/root:/bin/bash
root:x:11:11:root:/root:/bin/bash
root:x:12:12:root:/root:/bin/bash
root:x:13:13:root:/root:/bin/bash
root:x:14:14:root:/root:/bin/bash
root:x:15:15:root:/root:/bin/bash
root:x:16:16:root:/root:/bin/bash
root:x:17:17:root:/root:/bin/bash
root:x:18:18:root:/root:/bin/bash
root:x:19:19:root:/root:/bin/bash
root:x:20:20:root:/root:/bin/bash
root:x:21:21:root:/root:/bin/bash
root:x:22:22:root:/root:/bin/bash
root:x:23:23:root:/root:/bin/bash
root:x:24:24:root:/root:/bin/bash
root:x:25:25:root:/root:/bin/bash
root:x:26:26:root:/root:/bin/bash
root:x:27:27:root:/root:/bin/bash
root:x:28:28:root:/root:/bin/bash
root:x:29:29:root:/root:/bin/bash
root:x:30:30:root:/root:/bin/bash
root:x:31:31:root:/root:/bin/bash
root:x:32:32:root:/root:/bin/bash
root:x:33:33:root:/root:/bin/bash
root:x:34:34:root:/root:/bin/bash
root:x:35:35:root:/root:/bin/bash
root:x:36:36:root:/root:/bin/bash
root:x:37:37:root:/root:/bin/bash
root:x:38:38:root:/root:/bin/bash
root:x:39:39:root:/root:/bin/bash
root:x:40:40:root:/root:/bin/bash
root:x:41:41:root:/root:/bin/bash
root:x:42:42:root:/root:/bin/bash
root:x:43:43:root:/root:/bin/bash
root:x:44:44:root:/root:/bin/bash
root:x:45:45:root:/root:/bin/bash
root:x:46:46:root:/root:/bin/bash
root:x:47:47:root:/root:/bin/bash
root:x:48:48:root:/root:/bin/bash
root:x:49:49:root:/root:/bin/bash
root:x:50:50:root:/root:/bin/bash
root:x:51:51:root:/root:/bin/bash
root:x:52:52:root:/root:/bin/bash
root:x:53:53:root:/root:/bin/bash
root:x:54:54:root:/root:/bin/bash
root:x:55:55:root:/root:/bin/bash
root:x:56:56:root:/root:/bin/bash
root:x:57:57:root:/root:/bin/bash
root:x:58:58:root:/root:/bin/bash
root:x:59:59:root:/root:/bin/bash
root:x:60:60:root:/root:/bin/bash
root:x:61:61:root:/root:/bin/bash
root:x:62:62:root:/root:/bin/bash
root:x:63:63:root:/root:/bin/bash
root:x:64:64:root:/root:/bin/bash
root:x:65:65:root:/root:/bin/bash
root:x:66:66:root:/root:/bin/bash
root:x:67:67:root:/root:/bin/bash
root:x:68:68:root:/root:/bin/bash
root:x:69:69:root:/root:/bin/bash
root:x:70:70:root:/root:/bin/bash
root:x:71:71:root:/root:/bin/bash
root:x:72:72:root:/root:/bin/bash
root:x:73:73:root:/root:/bin/bash
root:x:74:74:root:/root:/bin/bash
root:x:75:75:root:/root:/bin/bash
root:x:76:76:root:/root:/bin/bash
root:x:77:77:root:/root:/bin/bash
root:x:78:78:root:/root:/bin/bash
root:x:79:79:root:/root:/bin/bash
root:x:80:80:root:/root:/bin/bash
root:x:81:81:root:/root:/bin/bash
root:x:82:82:root:/root:/bin/bash
root:x:83:83:root:/root:/bin/bash
root:x:84:84:root:/root:/bin/bash
root:x:85:85:root:/root:/bin/bash
root:x:86:86:root:/root:/bin/bash
root:x:87:87:root:/root:/bin/bash
root:x:88:88:root:/root:/bin/bash
root:x:89:89:root:/root:/bin/bash
root:x:90:90:root:/root:/bin/bash
root:x:91:91:root:/root:/bin/bash
root:x:92:92:root:/root:/bin/bash
root:x:93:93:root:/root:/bin/bash
root:x:94:94:root:/root:/bin/bash
root:x:95:95:root:/root:/bin/bash
root:x:96:96:root:/root:/bin/bash
root:x:97:97:root:/root:/bin/bash
root:x:98:98:root:/root:/bin/bash
root:x:99:99:root:/root:/bin/bash

```

Learning Outcomes

Upon completion of this course, you will be able to:

- 1 List the **objectives** and **functions** of modern Operating Systems.
- 2 Describe the concepts of **processes**, **threads**.
- 3 **Build** and **modify** Operating System **kernel** from source code.
- 4 Demonstrate **practical** experience under the Linux Operating System.
- 5 Construct **multi-tasking** programs and outline standard scheduling algorithms.

The screenshot shows a terminal window with two panes. The top pane displays the output of the 'cat /etc/passwd' command, showing system users like 'root:x:0:0:root:/root:/bin/bash' and regular users like 'daemon:x:65534:65534:daemon:/usr/sbin:/usr/sbin/nologin'. The bottom pane shows the output of the 'ps aux' command, listing running processes such as 'systemd', 'sshd', 'rsyslogd', and 'cron'.

Learning Outcomes

Upon completion of this course, you will be able to:

- 1 List the **objectives** and **functions** of modern Operating Systems.
- 2 Describe the concepts of **processes**, **threads**.
- 3 **Build** and **modify** Operating System **kernel** from source code.
- 4 Demonstrate **practical** experience under the Linux Operating System.
- 5 Construct **multi-tasking** programs and outline standard scheduling algorithms.
- 6 Develop pseudo-code implementation to deal with **deadlock**.

The screenshot shows a Linux terminal window with a dark background. The top part displays the system boot sequence, including the kernel version (Linux 2.6.32-041.el6.x86_64) and the root filesystem (rootfs). Below this, a process list is shown, listing various system processes such as init, kthreadd, ksoftirqd, and kworker. The bottom part of the terminal shows a list of running processes with columns for PID, PPID, USER, NAME, and STATE. The processes listed include init, kthreadd, ksoftirqd, kworker, and various user-space processes like sshd, rsyslogd, and cron.

Learning Outcomes

Upon completion of this course, you will be able to:

- 1 List the **objectives** and **functions** of modern Operating Systems.
- 2 Describe the concepts of **processes**, **threads**.
- 3 **Build** and **modify** Operating System **kernel** from source code.
- 4 Demonstrate **practical** experience under the Linux Operating System.
- 5 Construct **multi-tasking** programs and outline standard scheduling algorithms.
- 6 Develop pseudo-code implementation to deal with **deadlock**.
- 7 Describe and implement system **memory management** algorithms.

The screenshot shows a terminal window with a dark background. The top part displays the output of the 'cat /etc/passwd' command, showing system users like root, daemon, bin, and sys. Below that, the 'ps aux' command is executed, displaying a list of running processes with columns for PID, USER, PPID, STATUS, C, PRI, NI, ADDR, SZ, O, and COMMAND. The output shows various system processes like 'systemd', 'sshd', and 'rsyncd'.

Learning Outcomes

Upon completion of this course, you will be able to:

- 1 List the **objectives** and **functions** of modern Operating Systems.
- 2 Describe the concepts of **processes**, **threads**.
- 3 **Build** and **modify** Operating System **kernel** from source code.
- 4 Demonstrate **practical** experience under the Linux Operating System.
- 5 Construct **multi-tasking** programs and outline standard scheduling algorithms.
- 6 Develop pseudo-code implementation to deal with **deadlock**.
- 7 Describe and implement system **memory management** algorithms.
- 8 Describe **file systems** from **disk management** to high-level operations.

The screenshot shows a Linux terminal window with a dark background. The top part displays system information including the kernel version (Linux 2.6.32-0505.el6.x86_64), architecture (i686), and the root filesystem (xfs on /dev/sda1). Below this, the output of the 'top' command is visible, showing a list of running processes with columns for PID, USER, PR, NI, VIRT, RES, SHR, S, %CPU, %MEM, TIME+, and COMMAND. The 'top' command output is partially obscured by a window manager overlay.

- 1 Administrative Details
- 2 Course Description, Objectives, and Contents
- 3 Contents**
- 4 Examples of Operating Systems

Contents

- Computer Architecture (quick review)
- Operating System Introduction
- Tutorial on Linux

Contents

- Computer Architecture (quick review)
- Operating System Introduction
- Tutorial on Linux
- Process Concept (Process, PCB, operations on processes, IPC, etc.)
- Multithreaded-Programming (*POSIX Threads Programming*)
- Process Scheduling
- Synchronization: critical section problem, semaphores, etc.
- Deadlocks

Contents

- Computer Architecture (quick review)
- Operating System Introduction
- Tutorial on Linux
- Process Concept (Process, PCB, operations on processes, IPC, etc.)
- Multithreaded-Programming (*POSIX Threads Programming*)
- Process Scheduling
- Synchronization: critical section problem, semaphores, etc.
- Deadlocks
- Memory Management
- Virtual Memory

Contents

- Computer Architecture (quick review)
- Operating System Introduction
- Tutorial on Linux
- Process Concept (Process, PCB, operations on processes, IPC, etc.)
- Multithreaded-Programming (*POSIX Threads Programming*)
- Process Scheduling
- Synchronization: critical section problem, semaphores, etc.
- Deadlocks
- Memory Management
- Virtual Memory
- File and I/O Systems

Computer Architecture

- Computer elements (Processor, Memory, I/O, etc.)
- Program execution
- Interrupts

Operating System Introduction

- Operating System Structure
- System calls
- Multi-programming, time-sharing, job scheduling
- Physical and virtual Memory
- Installation and booting of an Operating System

Process Concept

A program in execution, which forms the basis of all computation

- Process Scheduling
- Operations on Processes (creating, deleting, etc.)
- Interprocess Communication IPC

Multithreaded Programming

A fundamental unit of CPU utilization that forms the basis of multithreaded computer systems

- Multithreading model overview
- POSIX Threads Programming

Scheduling

- Introduce process scheduling, which is the basis for multiprogrammed operating systems
- Describe various process-scheduling algorithms

Synchronization

- The critical-section problem (n processes all competing to use some shared data)
- Peterson's solution
- Synchronization hardware
- Semaphores
- Classic problem of synchronization

Deadlocks

- Several processes may compete for a finite number of **resources**.
 - 1 A requires printer and is in possession of scanner!
 - 2 B requires scanner and is in possession of printer!

Deadlocks

- Several processes may compete for a finite number of **resources**.
 - 1 A requires printer and is in possession of scanner!
 - 2 B requires scanner and is in possession of printer!
- That is, process A waits for process B **and** process B waits for process A

Deadlocks

- Several processes may compete for a finite number of **resources**.
 - 1 A requires printer and is in possession of scanner!
 - 2 B requires scanner and is in possession of printer!
- That is, process A waits for process B **and** process B waits for process A
- Discuss different methods for handling deadlocks:
 - Deadlock prevention
 - Deadlock avoidance
 - Deadlock detection
 - Recovery from deadlock

Memory Management

- We must keep several processes in memory.
- This drastically improve the speed of the computer!
- Various **memory-management** techniques:
 - Swapping
 - Segmentation
 - Paging

Virtual Memory

Virtual memory is a technique that allows the execution of a processes that are not completely in memory

- Describe the benefits of a virtual memory system
- Explain the concepts of demand paging, page-replacement algorithms, etc.

File and I/O systems

- File system provides mechanism for storage and access to file contents (including data and programs).
 - Explain the function of file systems
 - File-system design and implementation

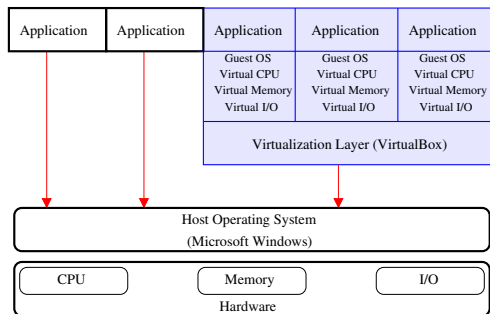
File and I/O systems

- File system provides mechanism for storage and access to file contents (including data and programs).
 - Explain the function of file systems
 - File-system design and implementation
- I/O vary widely in their function and speed (mouse, hard disk, etc.). Thus, varied method are needed to control them.
 - How to manage I/O operation and devises
 - Application I/O interface
 - Device drivers (if time permits)

- 1 Administrative Details
- 2 Course Description, Objectives, and Contents
- 3 Contents
- 4 **Examples of Operating Systems**

Popular Modern Operating Systems

- Examples: Linux (open source), Microsoft Windows, etc.
- During this course, we consider **Ubuntu Linux**. Ubuntu is popular Linux distribution.
- **Instructions for installing Ubuntu inside Windows or Mac OS X using *VirtualBox* on Moodle.** (Assignment zero)



History of Linux

- **Unix** (not free) was developed in 1969 by Bell Labs.

History of Linux

- **Unix** (not free) was developed in 1969 by Bell Labs.
- In 1984, Richard Stallman (researcher at MIT) created **GNU project** (a recursive acronym for “GNU’s Not Unix!”).
- GNU is **free and open-source** operating system that works like UNIX (same commands/API).

History of Linux

- **Unix** (not free) was developed in 1969 by Bell Labs.
- In 1984, Richard Stallman (researcher at MIT) created **GNU project** (a recursive acronym for “GNU’s Not Unix!”).
- GNU is **free and open-source** operating system that works like UNIX (same commands/API).

- **Mac OS X** is also based on UNIX. However, **Windows** is totally different.

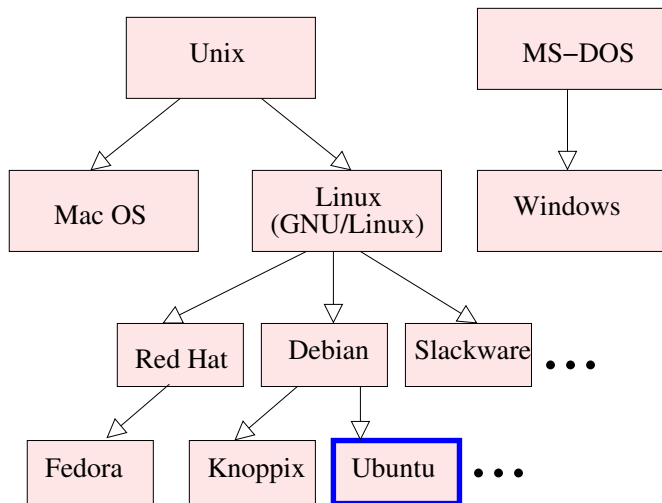
History of Linux

- **Unix** (not free) was developed in 1969 by Bell Labs.
- In 1984, Richard Stallman (researcher at MIT) created **GNU project** (a recursive acronym for “GNU’s Not Unix!”).
- GNU is **free and open-source** operating system that works like UNIX (same commands/API).
- **Mac OS X** is also based on UNIX. However, **Windows** is totally different.
- In 1991, **Linus Torvalds**, a student at Helsinki University (Finland), was partly influenced by a speech about GNU in Finland in 1990.
- Linus began creating his own operating system (Linux = **kernel**) during his free time.

History of Linux

- **Unix** (not free) was developed in 1969 by Bell Labs.
- In 1984, Richard Stallman (researcher at MIT) created **GNU project** (a recursive acronym for “GNU’s Not Unix!”).
- GNU is **free and open-source** operating system that works like UNIX (same commands/API).
- **Mac OS X** is also based on UNIX. However, **Windows** is totally different.
- In 1991, **Linus Torvalds**, a student at Helsinki University (Finland), was partly influenced by a speech about GNU in Finland in 1990.
- Linus began creating his own operating system (Linux = **kernel**) during his free time.
- GNU includes: gcc, gdb, bash shell, GNU Core Utilities (cat, ls, rm, cp, etc.).
- GNU + Linux = GNU/Linux = **Linux**.

Operating Systems



Kernel

The kernel is the core of the operating system:

- Interrupt handlers
- Scheduler to share processor
- Memory management
- File systems
- Inter-process communication
- Networking
- ...

Example of Linux kernel version (**major version** . **minor version** . **revision**).

Latest Stable Release:

4.12.9