

|  |
| --- |
| **Course Outline**  Department of Computing Science  Faculty of Science |

**COMP 3410 – 3 Credits**

**Operating Systems (3,1,0)**

**Winter 2018**

**Instructor:** Kevin O’Neil Phone/Voice Mail: 250-371-5789

E-Mail:koneil@tru.ca

Calendar Description

The purpose of this course is to provide students basic knowledge of operating systems, difference between the kernel and user modes, concepts of application program interfaces, methods and implementations of interrupts. Students are introduced to the schedulers, policies, processes, threads, memory management, virtual memory, protection, access control, and authentication. Students learn system calls in different popular operating systems used in the industry.

**Course/Learning Outcomes**

#### Upon successful completion of the course, the student will demonstrate the ability to:

#### Explain the objectives and functions of modern operating systems.

#### Describe how computing resources are used by application software and managed by system software.

#### Describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system. Evaluate the trade-offs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed.

#### Articulate the need for protection and security in an operating system

#### Prerequisites

A grade of C or better in COMP 2230: Data Structures and Algorithm Analysis

**Texts/Materials**

***Required Text:***

A. Silberschatz, P.B. Galvin, G. Gagne, *Operating System Concepts with Java*, John Wiley & Sons, 2009, 8th Ed.,

ISBN: 978-0470-509-494 (will be supplemented by selected materials on current operating systems)

***Recommended Text:***

* S. Tanenbaum, *Modern Operating Systems*, Pearson Prentice Hall, 4rd Ed., **2008**, ISBN-10: 013359162X • ISBN-13: 9780133591620

***Additional References (available in the library):***

* William Stallings, *Operating Systems (Internals and Design Principles)*, Pearson Prentice Hall, 6th Ed., **2009**,

ISBN: 0-13-601697-9.

* Gary Nutt, *Operating Systems,* Addison-Wesley, 3rd Ed., **2004**, ISBN: 0-201-77344-9.

## Student Evaluation

Unsupervised work 25%

Midterm 25%

Final Exam 50%

*Note:* Please note that in order to pass the course you must:

* Obtain an overall grade of at least 50%;
* Obtain a grade of at least 50% on the unsupervised work
* Obtain a grade of at least 50% on the supervised work

**If you fail to satisfy any of the above criteria, you will automatically fail the course.**

**Notes**

***Academic honesty:***

All work submitted must be your own. TRU's academic honesty are followed in this course—copied work will result in an automatic F for the course. The student is expected to be familiar with these policies--see the TRU Calendar.

***Assistance:***

All your instructors at TRU are here to help you succeed in each of your courses. I encourage you to come to see me if you need help. I would be delighted to see you.

***Attendance***

Since this is a programming course, there is a lot of programming involved. Attendance in every class is expected and very important for success in this course. The material presented in class will highlight important concepts and your assignments will build on those concepts to demonstrate practical applications of the material covered in class. Students who repeatedly miss lectures or seminars will be required to withdraw from the course.

***Class Conduct***

During lectures, the student is expected to act in a professional and respectful manner towards other students and instructors; students not conducting themselves in such a manner may be asked to leave. This courtesy is also expected during lab/seminar times, which are allotted only for course work.In particular, private conversations during lectures or labs are NOT allowed. Anyone doing so will be asked to leave the lecture or lab immediately.

***Late Homework Policy***

Assignments and Labs are due at the **BEGINNING** of the class/lab on the date indicated. Homework that is late shall receive a mark of zero (0).

**Course Coverage**

*Notes:* **1**. The operating system principles from chapters **1-13 and overview of the security and protection concepts (time permits) as implemented by OS** will be covered, supplemented by the additional lecture material.

**2.** Course will be a balanced combination of Assignments, Quizzes, Exercises, and Exams consisting of Theory, Programming, Research, Presentation, Team/Individual work.

**Syllabus – Lecture Topics  
Overview**

1. **Introduction** 
   1. Operating System functions
2. **Operating System Structures**
   1. System calls
   2. API
   3. Types of system calls
   4. Booting

**Process Management**

1. **Process** 
   1. Processes
   2. Inter Process Communication
   3. Examples of IPC systems
2. **Threads**
   1. Threads
   2. Thread types – user level thread /Kernel level thread etc.
3. **CPU Scheduling**
   1. Scheduling
   2. Algorithm evaluations
4. **Process Synchronization**
   1. Peterson’s Solution
   2. Mutex
   3. Semaphores
   4. Monitors
5. **Deadlocks** 
   1. Introduction
   2. Detection and recovery
   3. Avoidance
   4. Prevention

**Memory Management**

1. **Main Memory** 
   1. Main Memory
   2. Segmentation
   3. Paging
   4. Page tables
2. **Virtual Memory**
   1. Demand Paging
   2. Page replacement
   3. Allocation of frames

**Storage Management**

1. **File Systems Interface** 
   1. File System Interface
   2. Access methods
   3. Mounting concept
2. **File-system Implementation**
   1. File systems structure
   2. Allocation methods
   3. Free space management
3. **Mass storage structure** 
   1. Disk structure
   2. Disk scheduling
   3. Disk Management
   4. RAID structure
4. **Input / Output**
   1. I/O hardware
   2. DMA

**Protection and Security** **(if time permits)**

1. **Protection**
   1. Goals of protection
   2. Principles of protection
   3. Domain of Protection
   4. Access Matrix
   5. Implement of Access Matrix
   6. Access Control
2. **Security**
   1. The security problem
   2. Program threats
   3. System and Network threats
   4. Cryptography as a security tool
   5. User Authentication
   6. Implementing Security Defences
3. **Case study (if time permits)**
   1. Comparative Study of the latest operating systems **(Team Project)**
   2. Windows NT, 95, 98, 2000, ME, XP, Vista, 7
   3. Research Paper presentation

**Syllabus - Lab Topics**

|  |  |  |
| --- | --- | --- |
| Lab Topic | Tool | Duration in hours |
| Processes Ps, kill etc. | UNIX | 2 |
| IPC Introduction to threads | Java | 4 |
| Concurrency Control Monitors, Semaphores and locks | Java | 4 |
| Security features Permissions, sharing etc. | Unix/Windows | 2 |

**ACM / IEEE Knowledge Area Coverage**

**IEEE Knowledge Areas that contain topics and learning outcomes covered in the course**

|  |  |
| --- | --- |
| **Knowledge Area** | **Total Hours of Coverage** |
| Overview of OS | 3 |
| Operating System Principles | 3 |
| Concurrency | 6 |
| Scheduling and Dispatch | 4 |
| Memory Management | 4 |
| Virtual Machines | 4 |
| File Systems | 4 |
| Device Management | 4 |
| Security and Protection | 4 |

**IEEE Body of Knowledge coverage**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KA** | **Knowledge Unit** | **Topics Covered** | **T1 hour** | **T2 hour** | **Elective hours** |
|  | Overview | • Role and purpose of OS  • Functionality of a typical OS  • Mechanism to support client/server model, hand-held devices  • Design Issues  • Influence of security, networking, multimedia, windowing systems | 3 | 0 | 0 |
|  | Operating System Principles | • Structuring methods  • Abstraction, processes and resources  • API’s, System calls  • Interrupts : methods and implementations • Threads  • Kernel/User mode  • Device Organization | 3 | 0 | 0 |
|  | Concurrency | • States and state diagrams  • Process structures  • Dispatching and context switching  • The role of interrupts  • Managing atomic access to OS Objects  • Implementing synchronization primitives  • Multiprocessor issues (spin-lock, re-entrancy) | 6 | 0 | 0 |
|  | Scheduling and Dispatch | • Pre-emptive and non-preemptive scheduling  • Scheduler and policies  • Processes and threads  • Deadlines and real-time issues | 0 | 4 | 0 |
|  | Memory Management | • Review of physical memory and memory management hardware  • Working sets and thrashing  • Caching/ | 1 | 3 | 0 |
|  | Virtual Memory | • Types of Virtualization  • Paging and Virtual memory  • Virtual File System  • Hypervisors  • Portable Virtualizations  • Cost of Virtualization | 0 | 4 | 0 |
|  | File System | • Files : Data, metadata, sequential, random  • Directories : Content and structure  • File System : Structure, mount/unmount  • Standard implementing techniques  • memory mapped files  • naming, searching, access, backups  • journaling and log-structured file system | 0 | 4 | 0 |
|  | Device Management | • Serial and parallel devices  • Abstracting device differences  • Buffering strategies  • Direct Memory Access  • Recovery from Failure | 0 | 4 | 0 |
|  | Security and Protection | • Overview of system security  • Policy/mechanism separation  • Security methods and devices  • Protection, access control and authentication  • Backups | 0 | 4 | 0 |