CSCI 3753: Operating Systems

Final Exam Review

## Introduction

Operating System (OS):layer of software containing application programs with the users above it and machine below

* Convenience: transform raw hardware into a machine for users
* Efficiency: manages resources of the overall computer system

Operating systems are activated by interrupts from the hardware below or traps from software above

**Interrupts:** come from the hardware and are caused by devices requesting something from the CPU (processor)

**Traps:** come from the software and are caused by illegal events such as integer division by 0 or request from application programs or users. Also referred to as “software interrupts”

**System Calls:** a request to the OS from a program. A system call looks like a call to a C function and a set of calls looks like a library of predefined functions. Sent as traps or by a special system call instruction

The operating system responds to processes by handling system calls and error conditions.

The OS responds to devices by handling interrupts. True interrupts are from hardware.

Process**:** program in execution

OS manages the execution of programs using a table of processes. Each table entry contains the executing entity which has its own memory space. If two users are running the same program, the OS execution separates it by have one process for each user.

Services Provided by an OS:

* Process management and scheduling
* Main-memory management
* Secondary-memory management
* I/O System management (including interrupt handling)
* File management
* Protection & Security
* Networking
* Command interpretation

Nucleus:contains routines for managing registers, time, and handling device interrupts. It provides the environment in which processes exist. All processes depend on the services provided by the nucleus. Processes lie above the nucleus and devices below.

Contained in the nucleus:

**Interrupt handling:** interrupt service routines, usually one for each possible type of interrupt from the hardware

**Trap handling:** trap service routines, usually one for each type of trap from the processor

**Short Term Scheduling:** chooses which process to run next

**Process Management:** creates and deletes processes, assigns privileges and resources

**Interprocess Communication (IPC):** exchanges information between processes, communicates between them

Shell**:** interface between user and OS to transform a request from the user into a request to the OS.

Layered Design: series of software layers

Each layer of software provides services to the layers above and uses those provided by the layer below. A strongly layered approach only allows layers to use the services provided by the layer directly below it.

Advantages:

* Easy to design & implement one layer separately from others (modular)
* Easy to test and debug
* Easy to replace components

Disadvantages:

* Difficult to choose & define layers
* Slows OS

Virtual Machine: software emulation of a real (hardware) or imaginary machine completely implemented in software

User doesn’t have to purchase or maintain correct hardware if it can be emulated on another machine.

## Types of System Calls

Process Control:

* End/Abort
* Load/Execute
* Create/Terminate Process
* Get/Set Process Attributes
* Wait for Time
* Wait/Signal Event
* Allocate & Free Memory

File Management:

* Create/Delete File
* Open/Close File
* Read/Write/Reposition File
* Get/Set File Attributes

Device Management:

* Request/Release Devices
* Read/Write/Reposition Devices
* Get Device Attributes
* Logically Attach/Detach Devices

Information Maintenance:

* Get/Set Time or date
* Get/Set System Data
* Get/Set Process/File/Device Attributes

Communications:

* Create/Delete Communication Connection
* Send/Receive Messages
* Transfer Status Information
* Attach/Detach Remote Devices

## File Systems, Logical View

## Nucleus and Processes

## UNIX System Calls

## Scheduling

## Process Synchronization and Deadlocks

## Memory Management

## Files Systems, Physical View (Disk Allocation)

## Input/Output and Disk Head Scheduling

## Distributed Algorithms