

# **CS 350: Operating Systems**

Charles Shen

Fall 2016, University of Waterloo

Notes written from Gregor Richards's lectures.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Application View of an Operating System . . . . .	1
1.2	System View of an Operating System . . . . .	1
1.3	Implementation View of an Operating System . . . . .	2
1.4	Operating System Abstractions . . . . .	2
<b>2</b>	<b>Threads and Concurrency</b>	<b>3</b>
2.1	OS/161's Thread Interface . . . . .	3
<b>3</b>	<b>Processes and System Calls</b>	<b>4</b>
<b>4</b>	<b>Assignment 2A Review</b>	<b>5</b>
<b>5</b>	<b>Virtual Memory</b>	<b>6</b>
<b>6</b>	<b>Scheduling</b>	<b>7</b>
<b>7</b>	<b>Devices and Device Management</b>	<b>8</b>
<b>8</b>	<b>File Systems</b>	<b>9</b>
<b>9</b>	<b>Interprocess Communications and Networking</b>	<b>10</b>
	<b>Indices</b>	<b>11</b>

# 1 Introduction

There are three views of an operating system:

1. **Application View** (Section 1.1): what service does it provide?
2. **System View** (Section 1.2): what problems does it solve?
3. **Implementation View** (Section 1.3): how is it built?

*An operating system is part cop, part facilitator.*

**kernel:** The operating system kernel is the part of the operating system that responds to system calls, interrupts and exception.

**operating system (OS):** The operating system as a whole includes the kernel, and may include other related programs that provide services for application such as utility programs, command interpreters, and programming libraries.

## 1.1 Application View of an Operating System

The OS provides an execution environment for running programs.

- The execution environment provides a program with the processor time and memory space that it needs to run.
- The execution environment provides interfaces through which a program can use networks, storage, I/O devices, and other system hardware components. Interfaces provide a simplified, abstract view of hardware to application programs.
- The execution environment isolates running programs from one another and prevents undesirable interactions among them.

## 1.2 System View of an Operating System

The OS manages the hardware resources of a computer system.

- Resources include processors, memory, disks and other storage devices, network interfaces, I/O devices such as keyboards, mice and monitors, and so on.
- The operating system allocates resources among running programs. It controls the sharing of resources among programs.

- The OS itself also uses resources, which it must share with application programs.

### 1.3 Implementation View of an Operating System

The OS is a concurrent, real-time program.

- Concurrency arises naturally in an OS when it supports concurrent applications, and because it must interact directly with the hardware.
- Hardware interactions also impose timing constraints.

### 1.4 Operating System Abstractions

The execution environment provided by the OS includes a variety of abstract entities that can be manipulated by a running program.

Examples:

- **files and file systems:** abstract view of secondary storage
- **address spaces:** abstract view of primary memory
- **processes, threads:** abstract view of program execution
- **sockets, pipes:** abstract view of network or other message channels

## 2 Threads and Concurrency

Threads provide a way for programmers to express *concurrency* in a program.

A normal *sequential program* consists of a single thread of execution.

In threaded concurrent programs, there are multiple threads of executions that are all occurring at the same time.

### 2.1 OS/161's Thread Interface

Create a new thread:

```
int thread_fork(  
    const char *name,           // name of new thread  
    struct proc *proc,         // thread's process  
    void (*func)                // new thread's function  
        (void *, unsigned long),  
    void *data1,                // function's first param  
    unsigned long data2         // function's second param  
);
```

Terminating the calling thread:

```
void thread_exit(void);
```

Voluntarily yield execution:

```
void thread_yield(void);
```

### **3 Processes and System Calls**

## **4 Assignment 2A Review**

## 5 Virtual Memory



## 6 Scheduling

## **7 Devices and Device Management**

## 8 File Systems

## **9 Interprocess Communications and Networking**

## Indices

Application View, [1](#)

Implementation View, [1](#)

kernel, [1](#)

operating system, [1](#)

System View, [1](#)