

COMPSCI 340 and SOFTENG 370

Operating Systems

Lecturers

Robert Sheehan (r.sheehan@auckland.ac.nz)

Rm 303.571

office hours – most times Monday, Wednesday, Friday

Tutors

Richard Priest (rpri032@ec.auckland.ac.nz)

Martin Tingstad (mtin010@ec.auckland.ac.nz)

Textbook

Operating System Concepts with Java – Silberschatz, Galvin and Gagne

See the *resources* page of the web site for the programs included in the text.

Test (10%)

Tuesday 23rd of August

Exam (70%)

November?

Three Assignments (20%)

Web site - www.cs.auckland.ac.nz/compsci340sc

Course Wiki - wiki.cs.auckland.ac.nz/os

You have to pass the assignments and exams separately, and get an overall pass.

Assignment pass grade 0%.

Tutorials

Monday 11am, 3pm

Wednesday 11am, 4pm

Friday 11am, 1pm

Start 25th of July

Not compulsory

Sign up sheets outside Robert's office from Tuesday
19th of July to Friday the 22nd of July.

First come, first served.

Rooms to be announced.

What is an Operating System?

Examples

MacOS X

Windows XP

Linux

UNIX

Plan9

Amoeba

VM/CMS

The software which makes the computer usable.

It is impossible to use modern machines without an OS.

The collection of software sold (or freely available) as an OS.

Are these things part of the OS?

file system

communication system

process manager

security manager

memory manager

graphical user interface

backup system

Web browser

video player

compiler

Java environment

Extreme approaches

Minimalist understanding

OS software is the minimum amount of software required to allow the computer to function.

kernel – usually in memory always

process/thread management

communications

memory management

file management

monolithic and micro-kernels.

Maximalist understanding

All the software which comes with a standard release of the OS.

many utilities and programs

Usable vs Efficient

Some OSs are designed for specific needs

- factory control systems
- aircraft control
- database servers
- PDA's

Others are general purpose

- desktop computers

A trade off between usability and efficiency.

Usable – for whom?

- the developer of the system
- a computer scientist
- a data entry operator
- a child
- a person with a disability
- an “ordinary” user

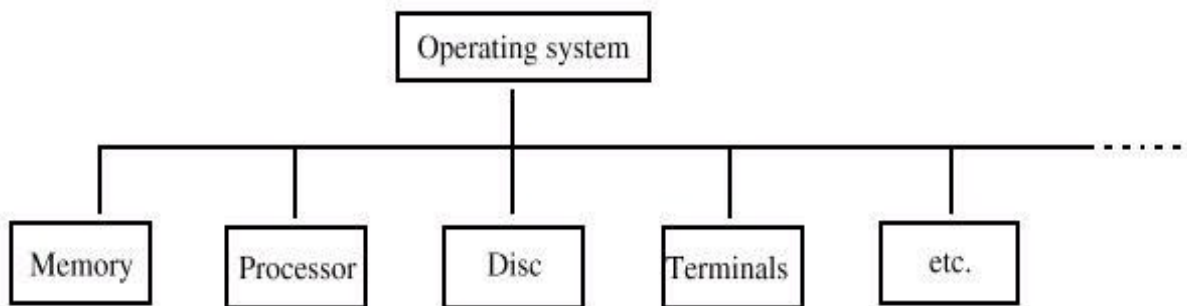
Efficient

- real-time systems
- dealing with thousands of transactions a second

OS Themes

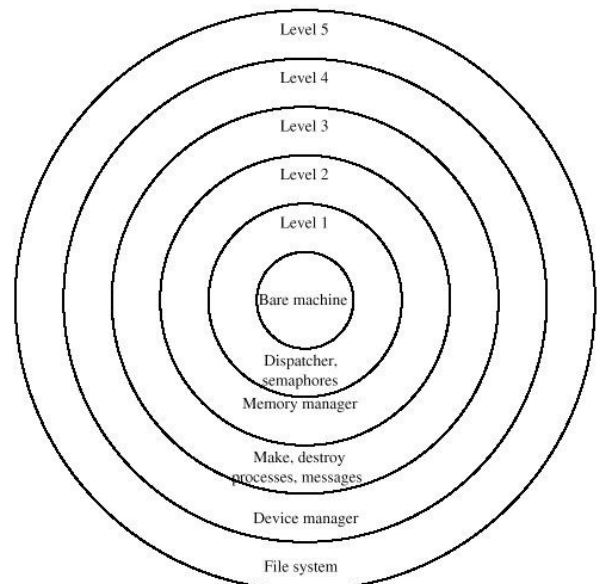
Manager model

The OS is a collection of managers.
It prevents improper use of devices.
Each manager is independent
and maintains tables of information

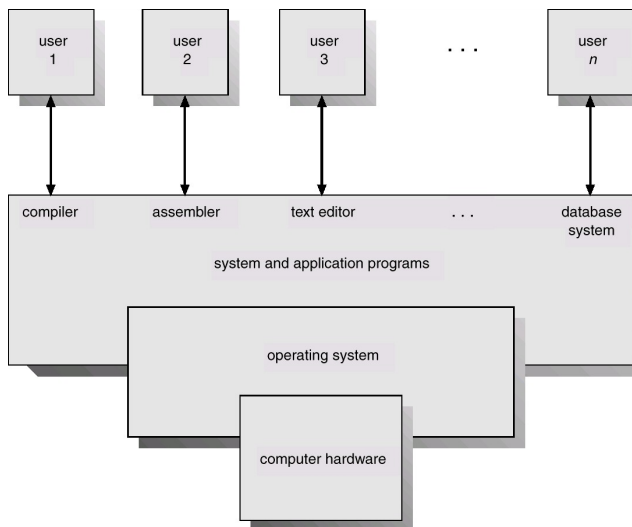


Onion model

The OS is a series of layers.
Outer layers can access resources contained in inner
layers.
But not vice-versa.



OS Themes (cont.)



Resource allocator model

Related to the manager model.

The emphasis is on providing the services programs need.

Must be fair. (Whatever that means.)

Dustbin model

This sees the OS as all the bits no one else wants to do.

Getting work done model

We only use computers to do something else:

- write an essay

- calculate a mortgage repayment

- find information

- download a song

- play a game

The OS has to help us to get our work done.

Things I expect you to know.

From 210/252 you should have some idea of

- interrupts
- security & protection
- file systems
- virtual memory
- processes and threads
- a Unix shell

I encourage everyone to install Linux.

You learn a lot by going through the process.

But please backup any important data first.

See www.cs.auckland.ac.nz/compsci340s2c/resources/
for further information.

Java

The textbook uses Java to illustrate algorithms and OS concepts

If you are new to Java read *Appendix A – JAVA PRIMER*.

Since Java is designed to run on a large number of different OSs it includes many OS concepts within the virtual machine.

- Security

- Networking

- Distributed computing

- Threads

- Synchronization

- Streams and filters

- Exception handling

Things you should know by the end.

You should be able to discuss questions like these intelligently by the end of the course.

How does typing a key cause a character to appear in a window on the display?

How does your computer keep several applications running at once?

How do remote files look as though they are local?

How does processing get distributed over the most appropriate computers on a network?

How does the failure of a computing resource get handled to minimize the disruption to the wider system?

Ruby

Why is a language like Ruby useful for an OS course?

It gives easy access to Unix system calls

e.g. `fork` and `exec`

and the results of commands

e.g.

```
result = open("| ps")  
puts result.readlines  
result.close
```

It has powerful regular expression support.

It has built-in exception handling.

Running Ruby

Use any text editor to type your Ruby programs.

Normally Ruby programs end with `*.rb`

```
ruby example.rb
```

Also try interactive Ruby

```
irb
```

And find out about Ruby calls with

```
ri
```

A little bit of Ruby

Ruby is a dynamically typed language.

You don't declare variables you just use them
e.g. (try the following in irb)

```
a = 5
```

```
a.next # everything is an object
```

```
a
```

```
b = [1, "two", 1+2] # an array
```

```
b << "four" # appending to an array
```

```
b[6] = 7 # assigning beyond an array
```

```
b
```

```
c = {"number"=>1, "colour"=>"blue",  
     "list"=>[1,4, 9, 16]} # a hash
```

```
c["colour"]
```

```
def squares(n)  
  for i in 1..n  
    puts i*i  
  end  
end
```

```
squares(10)
```

```
# Constants - including classnames, start with  
  a capital letter.
```

```
# Instance variables start with "@"
```

Before the next lecture

Textbook

Read Chapter 1.1

Ruby

Read the “Ruby.new” chapter in the online
“Programming Ruby” book.

http://www.cs.auckland.ac.nz/references/ruby/doc_bundle/ProgrammingRuby

Start writing on the Wiki.