Introduction

Why Study Operating Systems?

- Understand the model of operation
  - Easier to see how to use the system
  - Enables you to write efficient code
- Learn to design an OS
- Even so, OS is pure overhead of real work
- Application programs have the real value to person who buys the computer

Perspectives of the Computer

System Software

- Independent of individual applications, but common to all of them
- Examples
  - C library functions
  - A window system
  - A database management system
  - Resource management functions
  - The OS and kernel calls

Using the System Software

Application Software, System Software, and the OS
The OS as Resource Manager

- **Process**: An executing program
- **Resource**: Anything that is needed for a process to run
  - Memory
  - Space on a disk
  - The CPU
- “An OS creates resource abstractions”
- “An OS manages resource sharing”

Resource Abstraction

```c
load(block, length, device);
seek(device, 236);
out(device, 9);

write(char *block, int len, int device, int track, int sector) {
  ...
  load(block, length, device);
  seek(device, 236);
  out(device, 9);
  }

write(char *block, int len, int device, int addr);

print("Python is at this level")
```

Disk Abstractions

- Direct Control
- `write()` abstraction
- `fprintf()` abstraction

Abstract Resources

- User Interface
- Application
- Abstract Resources (API)
- Middleware
- OS Resources (OS Interface)
- OS
- Hardware Resources

Abstract Machines

- Idea
- Program
- Abstract Machines
- Physical Machine
- Result

Resource Sharing

- Space- vs time-multiplexed sharing
- To control sharing, must be able to isolate resources
- OS usually provides mechanism to isolate, then selectively allows sharing
  - How to isolate resources
  - How to be sure that sharing is acceptable
- Concurrency
The OS as a Conductor

The OS coordinates the sharing and use of all the components in the computer.

Multiprogramming

- Technique for sharing the CPU among runnable processes
  - Process may be blocked on I/O
  - Process may be blocked waiting for other resource, including the CPU
- While one process is blocked, another might be able to run
- Multiprogramming OS accomplishes CPU sharing “automatically” – scheduling
- Reduces time to run all processes

Multiprogramming (2)

How Multiprogramming Works

Speeding Up the Car Wash

(a) The Sequential Car Wash

(b) The Parallel Car Wash

Multiprogramming Performance

(a) P’s Use of Machine Resources

(b) All Processes’ Use of Machine Resources
OS Strategies

- Batch processing
- Timesharing
- Personal computer & workstations
- Process control & real-time
- Network
- Distributed
- Small computers

Batch Processing

- Uses multiprogramming
- Job (file of OS commands) prepared offline
- Batch of jobs given to OS at one time
- OS processes jobs one-at-a-time
- No human-computer interaction
- OS optimizes resource utilization
- Batch processing (as an option) still used today

A Shell Script Batch File

```
cc -g -c menu.c
driver < test_data > test_out
lpr -PthePrinter test_out
```

A Batch File for the Cluster

```
#!/bin/sh

#PBS -N namd
#PBS -l nodes=2:ppn=2
#PBS -l cput=2:00
#PBS -m be

echo "The nodefile is ${PBS_NODEFILE} and it contains:

cat ${PBS_NODEFILE}

MODELFILE=${pbs_nodefile}

for x in "cat ${PBS_NODEFILE} | sort -u" ; do
  echo ${x} >> ${MODELFILE};
done

echo MODELFILE is:
cat ${MODELFILE}

time mpirun -machinefile ${PBS_NODEFILE} -np ${PBS_NUMPROC} /usr/local/src/NAMD_2.6_Source/Linux- amd64-MPI/namd -np ${PBS_NUMPROC} --nodelist ${PBS_NODEFILE} -alanin"

Timesharing Systems
Timesharing Systems(2)

- Uses multiprogramming
- Support interactive computing model (Illusion of multiple consoles)
- Different scheduling & memory allocation strategies than batch
- Tends to propagate processes
- Considerable attention to resource isolation (security & protection)
- Tend to optimize response time

Personal Computers

- CPU sharing among one person's processes
- Power of computing for personal tasks
  - Graphics
  - Multimedia
- Trend toward very small OS – Android, Windows RT, Blackberry, LG TV!
- OS focus on resource abstraction
- Rapidly evolved to “personal multitasking” systems

Process Control & Real-Time

- Computer is dedicated to a single purpose
- Classic embedded system
- Must respond to external stimuli in fixed time
- Continuous media popularizing real-time techniques
- An area of growing interest

Networks

- LAN (Local Area Network) evolution
- High speed communication means new way to do computing
  - Shared files
  - Shared memory
  - Shared procedures/objects
  - ???

Distributed OS

- Wave of the future? Cloud computing?

Small Computers

- PDAs, STBs, embedded systems became commercially significant
- Have an OS, but
  - Not general purpose
  - Limited hardware resources
  - Different kinds of devices
    - Touch screen, no keyboard
    - Evolving & leading to new class of OSes
- PalmOS, Pocket PC (WinCE), VxWorks, Android,…
Evolution of Modern OS

Examples of Modern OS

The Microsoft OS Family

Microsoft Windows NT

Windows NT (cont)

VxWorks Organization
## Windows CE Organization

<table>
<thead>
<tr>
<th>Shells and Applications</th>
<th>Win32 API (&amp; Network Extensions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td>Object Store</td>
</tr>
<tr>
<td></td>
<td>Network and Comm Services</td>
</tr>
<tr>
<td></td>
<td>Graphics, Window Mgr, Event Mgr</td>
</tr>
<tr>
<td>Device Drivers</td>
<td>OEM Abstraction Layer</td>
</tr>
</tbody>
</table>

## Homework recommendations

- Read Chapter 1
- Begin reading Chapter 2
- Work *all* problems in Chapter 1
- Ask questions about problems that you aren't comfortable with!