CSC 360
Operating Systems
Overview

Wenjun Yang

https://wenjun-y.github.io/csc360

Summer 2026

What is an Operating System?

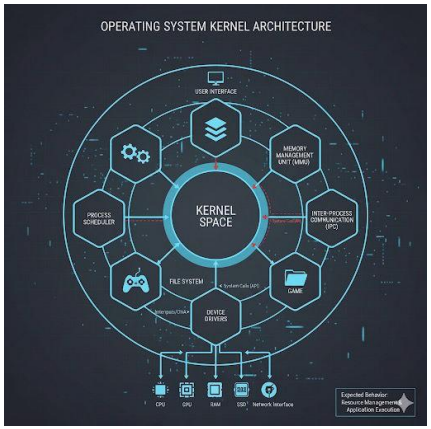
What's operating system?

Operating

System

Manages multiple tasks and users

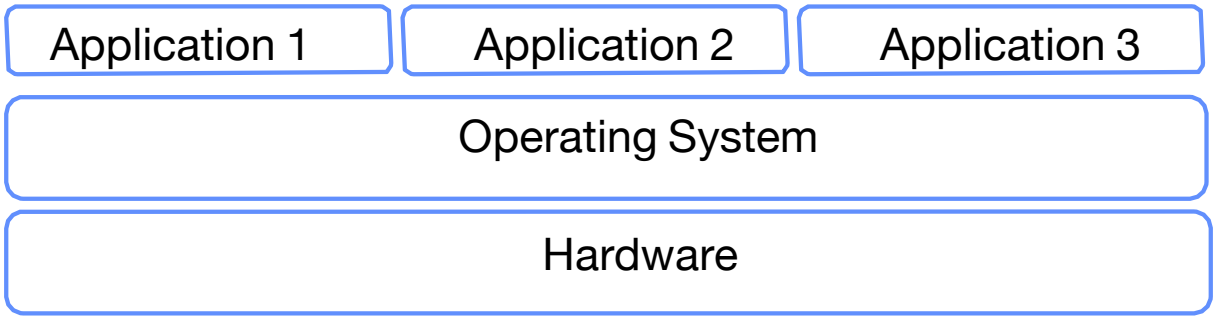
A set of interconnected components with an expected behavior observed at the interface with its environment



Generated by Gemini 2.5

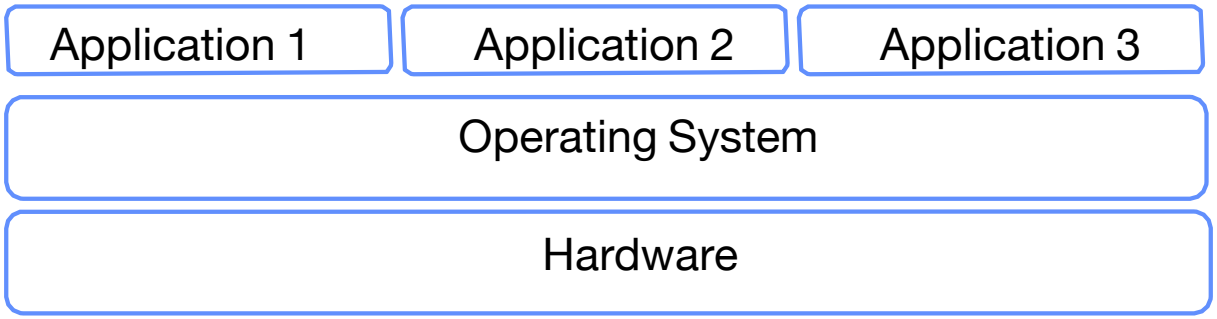
Operating System (v1)

An operating system is the layer of software that interfaces (many) applications running on a machine with (diverse) hardware resources of that machine



Operating System (v2)

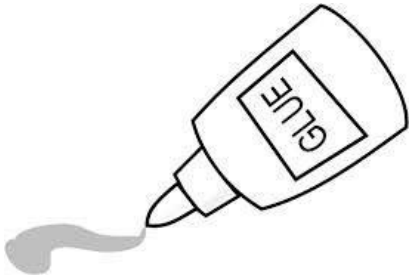
An operating system implements a virtual machine for the application whose interface is more convenient than the raw hardware interface (convenient = portability, reliability, security)



Roles of OS

Referee
Manage protection, isolation, and sharing of resources

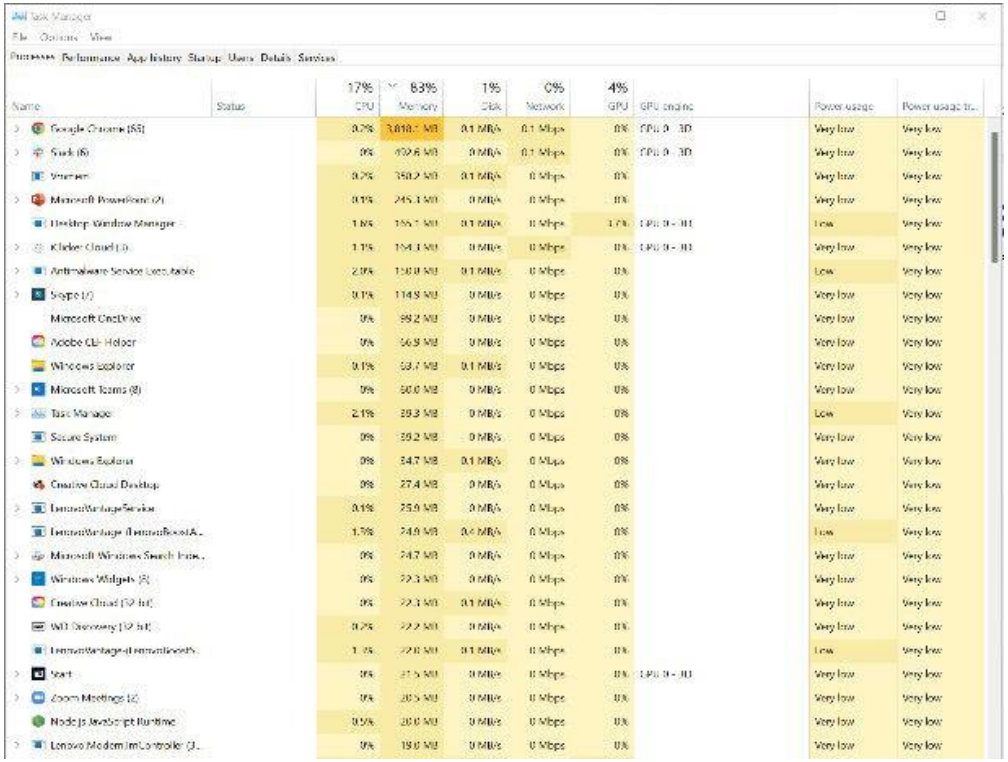
Illusionist
Provide clean, easy-to-use abstractions of physical resources



Glue
Provides a set of common services

OS as a referee

Allow multiple (untrusted) applications to run concurrently



OS as a referee

Fault Isolation

Isolate programs from each other

Isolate OS from other programs

Process

Dual Mode Execution

Resource Sharing

How to choose which task to run next?

How to split physical resources?

Scheduling

Communication

How can OS support communication to share results?

Pipes/Sockets

What does this program do?

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <assert.h>

int main(int argc, char *argv[]){
  char *str = argv[1];
  while (1) {
    printf("%s\n", str);
  }
  return 0;
}
```

```
ion@laptop> gcc -o cpu cpu.c -Wall
ion@laptop> ./cpu A
A
A
A
A
…
```

```
ion@laptop> ./cpu A & ./cpu B & ./cpu C
```

a)

```
A
A
…
```

b)

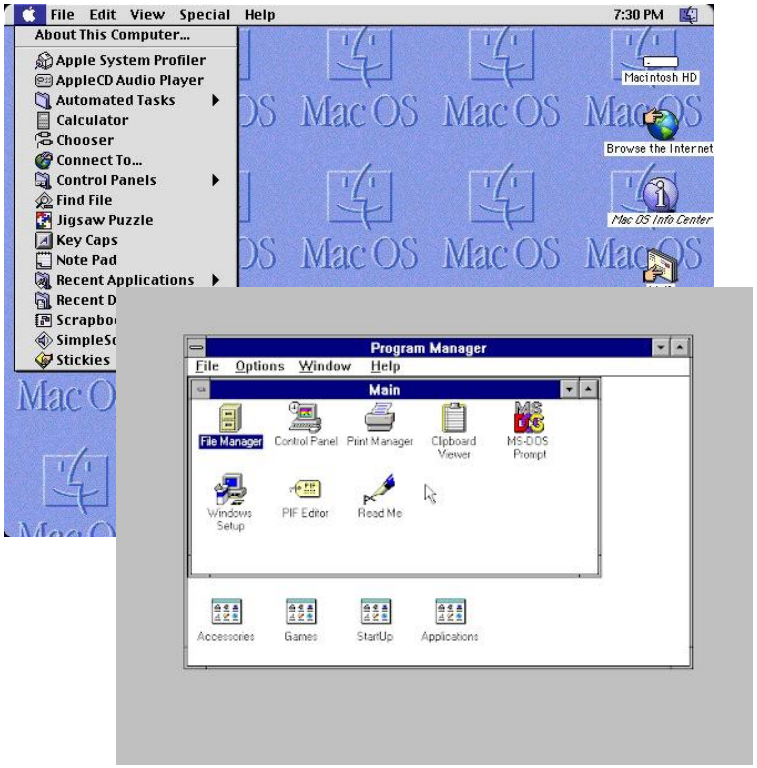
```
C
A
B
C
…
```

c)

```
B
A
C
B
C
…
```

```
ion@laptop> ./cpu & ; ./cpu B
Segmentation Fault
B
…
```

Refereeing is hard!



Up to MacOS 9x and Windows 3.1

OS cannot force program to give up control!

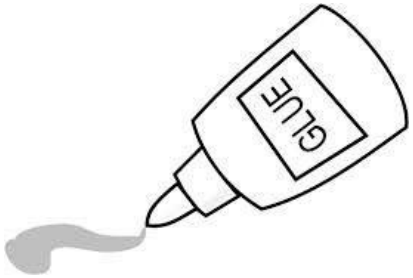
```
ion@very-old-laptop>
./cpu A & ./cpu B & ./cpu C

A            A            A            A            A
A A
...
```

Three main roles

Referee
Manage protection, isolation, and sharing of resources

Illusionist
Provide clean, easy-to-use abstractions of physical resources



Glue
Provides a set of common services

OS as Illusionist

Mask the restrictions inherent in computer hardware through virtualization

All alone

Provide illusion that application has exclusive use of resources

All powerful

Provide illusion that hardware resources are infinite

All expressive

Provide illusion of hardware capabilities that are not physically present

What does this program do?

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]){
  int *p = malloc(sizeof(int));
  printf("(%d) p: %p\n", getpid(), p);
  *p = 0;
  while (1) {
    *p = *p + 1;
    printf("(%d) p: %d\n", getpid(), *p);
  }
  return 0;
}
```

```
ion@laptop> gcc -o memory memory.c -Wall
ion@laptop> ./memory
(120) p: 0x200000
(120) p: 1
(120) p: 2
(120) p: 3
(120) p: 4
ion@laptop> ./memory & ./memory
(120) p: 0x200000
(254) p: 0x200000
```

a)

```
(120) p: 1
(254) p: 2
(120) p: 3
(254) p: 4
(120) p: 5
(254) p: 6
…
```

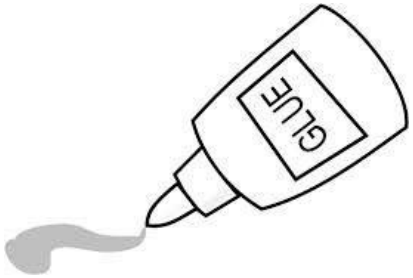
b)

```
(120) p: 1
(254) p: 1
(120) p: 2
(254) p: 2
(120) p: 3
(254) p: 3
…
```

Three main roles

Referee
Manage protection, isolation, and sharing of resources

Illusionist
Provide clean, easy-to-use abstractions of physical resources



Glue
Provides a set of common services

OS as Glue

Provide set of common, standard services to applications to simplify and regularize their design

Make sharing easier

Simpler if all assume same basic primitives

Maximise reuse

Avoid re-implementing functionality from scratch.

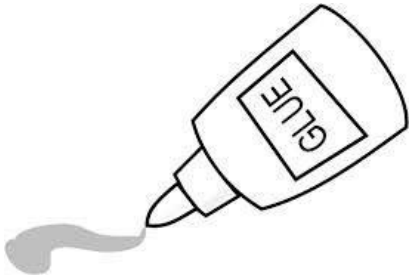
Evolve components independently

File System, User Interface, Network, etc.

Three main roles

Referee
Manage protection, isolation, and sharing of resources

Illusionist
Provide clean, easy-to-use abstractions of physical resources

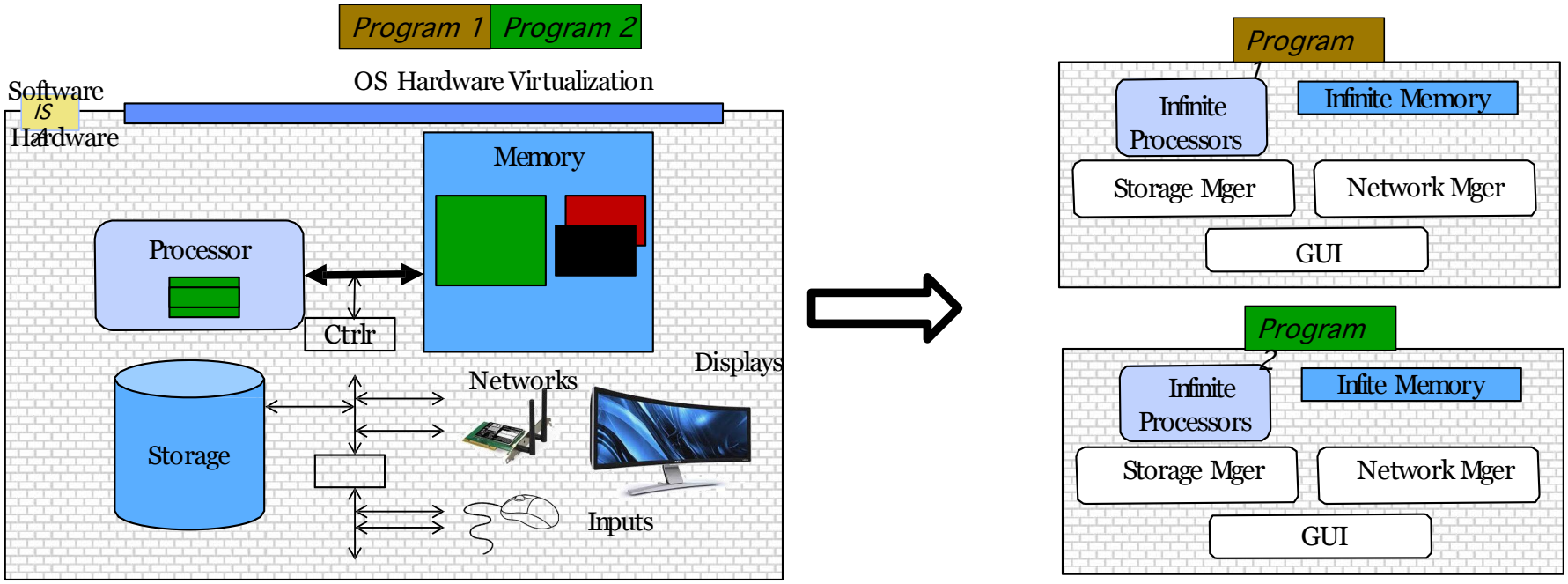


Glue
Provides a set of common services

Put all together

Referee + illusionist + Glue

=> Easy to use virtual machine



This lecture so far

- Summary of goals
  - Why should you care?
    - The OS is everywhere
  - Why is it hard?
    - Deal with many different devices, many different time scales. Safety-critical
  - What is an Operating System?
    - Provides abstraction of a simple, infinite virtual machine
    - Three roles: illusionist, referee and glue
    - A good OS cares about performance, reliability, security and portability

Next lecture

- Interfaces to OS
  - CLI, GUI, system calls, API
  - read OSC7 Chapter 2 (or OSC6 Chapter 3)