

CSC 360

Operating Systems

Overview

Wenjun Yang

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

Summer 2026

What is an Operating System?

What's operating system?

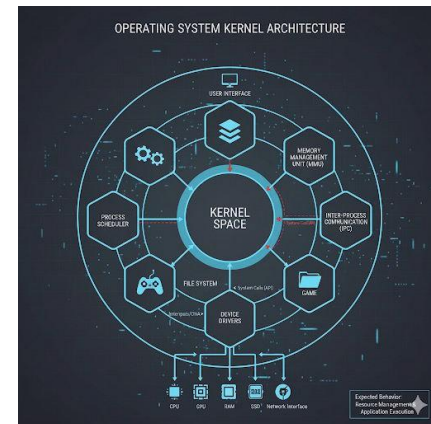
Operating

Manages multiple tasks and users



System

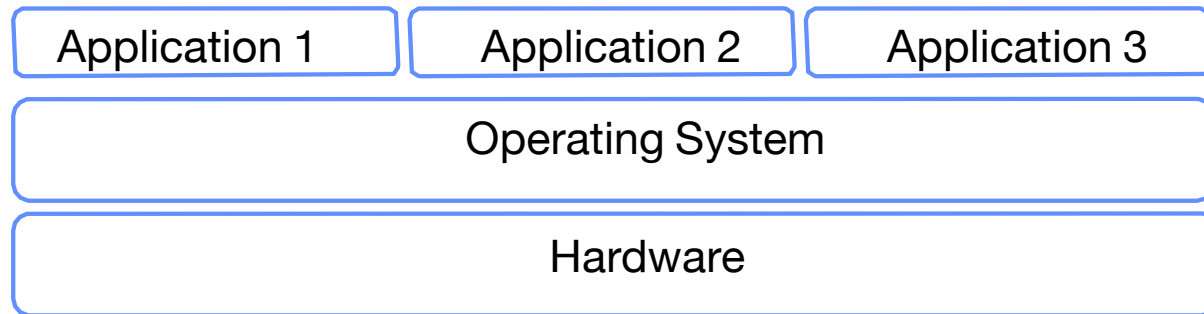
A set of interconnected components with an expected behavior observed at the interface with its environment



Generated by Gemini 2.5

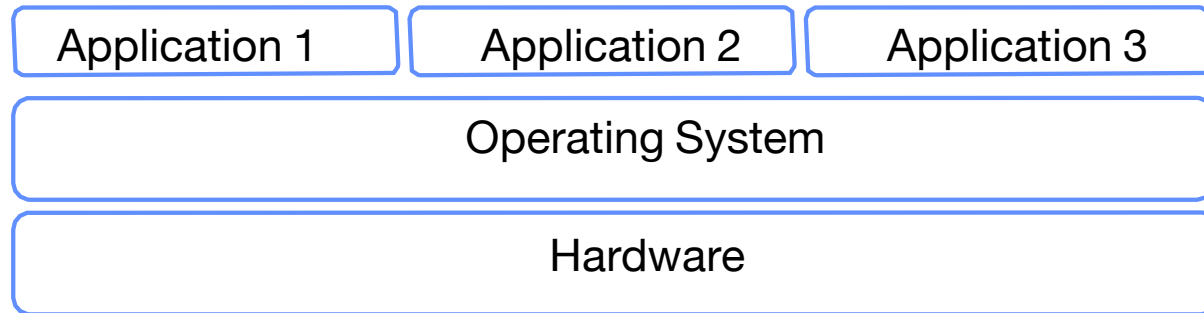
Operating System (v1)

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



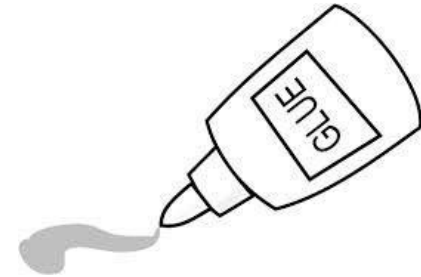
Referee

Manage protection, isolation, and sharing of resources



Illusionist

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

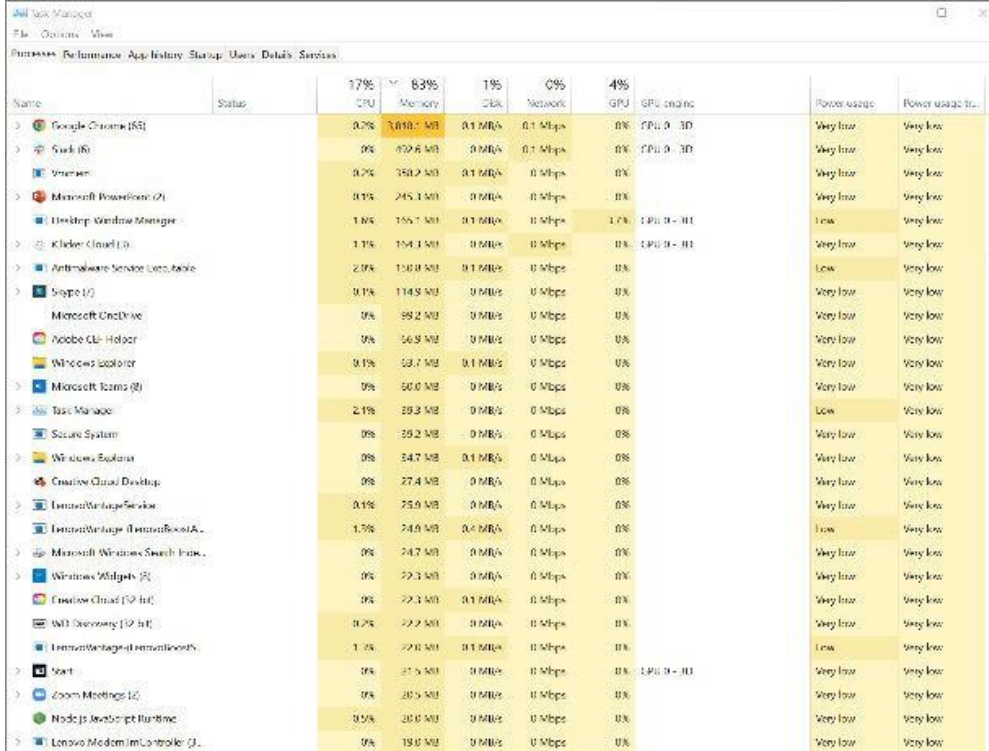


Glue

Provides a set of common services

OS as a referee

Allow multiple (untrusted) applications to run concurrently



The screenshot shows the Windows Task Manager Performance tab. At the top, system resource usage is summarized: CPU 17%, Memory 83%, Disk 1%, Network 0%, and GPU 4%. Below this is a table listing running applications and their resource usage.

Name	Status	CPU	Memory	Disk	Network	GPU	GPU end use	Power usage	Power usage fr.
Google Chrome (65)		0.2%	3,018 MB	0.1 MB/s	0.1 Mbps	0%	GPU 0 - 3D	Very low	Very low
Task (6)		0%	429.6 MB	0 MB/s	0.1 Mbps	0%	GPU 0 - 3D	Very low	Very low
System		0.2%	348.2 MB	0.1 MB/s	0 Mbps	0%		Very low	Very low
Microsoft PowerPoint (2)		0.1%	245.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Desktop Window Manager		1.8%	188.7 MB	0.1 MB/s	0 Mbps	1.7%	GPU 0 - 3D	Low	Very low
Kleider Cloud (3)		1.1%	154.3 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D	Very low	Very low
Antivirus Service Updatable		2.0%	130.8 MB	0.1 MB/s	0 Mbps	0%		Low	Very low
Slingshot (2)		0.1%	114.9 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Microsoft OneDrive		0%	99.2 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Google Chrome Helper		0%	56.9 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Windows Explorer		0.1%	53.7 MB	0.1 MB/s	0 Mbps	0%		Very low	Very low
Microsoft Teams (8)		0%	50.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Task Manager		2.1%	49.3 MB	0 MB/s	0 Mbps	0%		Low	Very low
Secure System		0%	39.2 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Windows Explorer		0%	34.7 MB	0.1 MB/s	0 Mbps	0%		Very low	Very low
Creative Cloud Desktop		0%	27.4 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Internet Widgets Service		0.1%	25.9 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Internet Widgets (InternetExeSt...		1.5%	24.9 MB	0.1 MB/s	0 Mbps	0%		Low	Very low
Microsoft Windows Search Index...		0%	24.7 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Windows Widgets (2)		0%	22.3 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Creative Cloud (39 bit)		0%	22.1 MB	0.1 MB/s	0 Mbps	0%		Very low	Very low
Win (Discovery) (2 bit)		0.2%	22.2 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Internet Widgets (InternetExeSt...		1.2%	22.0 MB	0.1 MB/s	0 Mbps	0%		Low	Very low
Start		0%	21.5 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D	Very low	Very low
Zoom Meetings (2)		0%	20.5 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Node.js lewdscript Runtime		0.5%	16.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Lenovo Modem (m) Controller (2)		0%	19.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low

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
```

```
AA
```

```
...
```

Three main roles



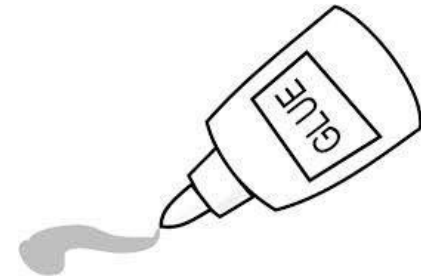
Referee

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



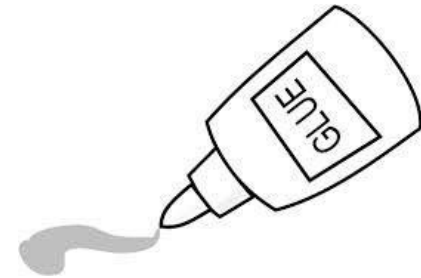
Referee

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



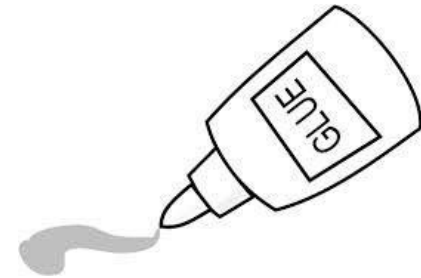
Referee

Manage protection, isolation, and sharing of resources



Illusionist

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

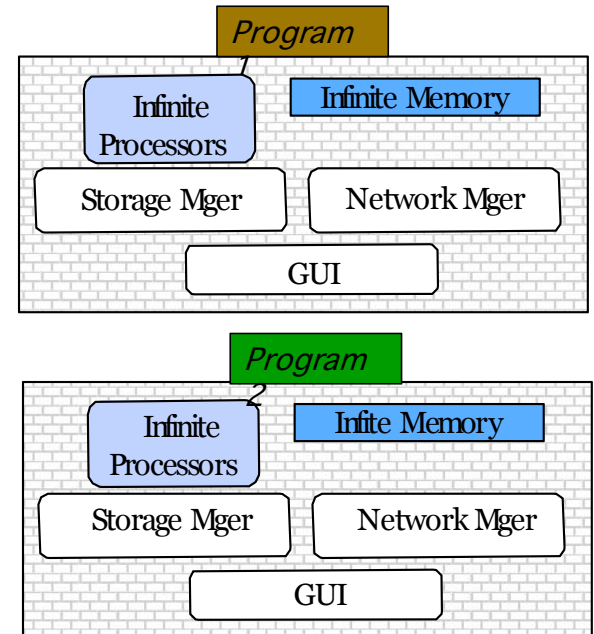
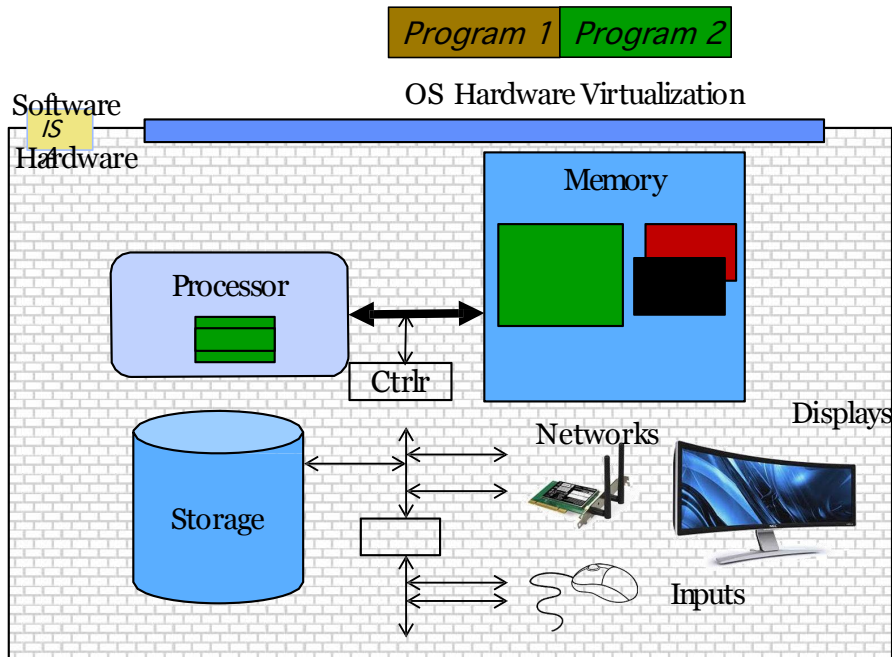


Glue

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)