Introduction to the OS

Gerald S. Tompkins

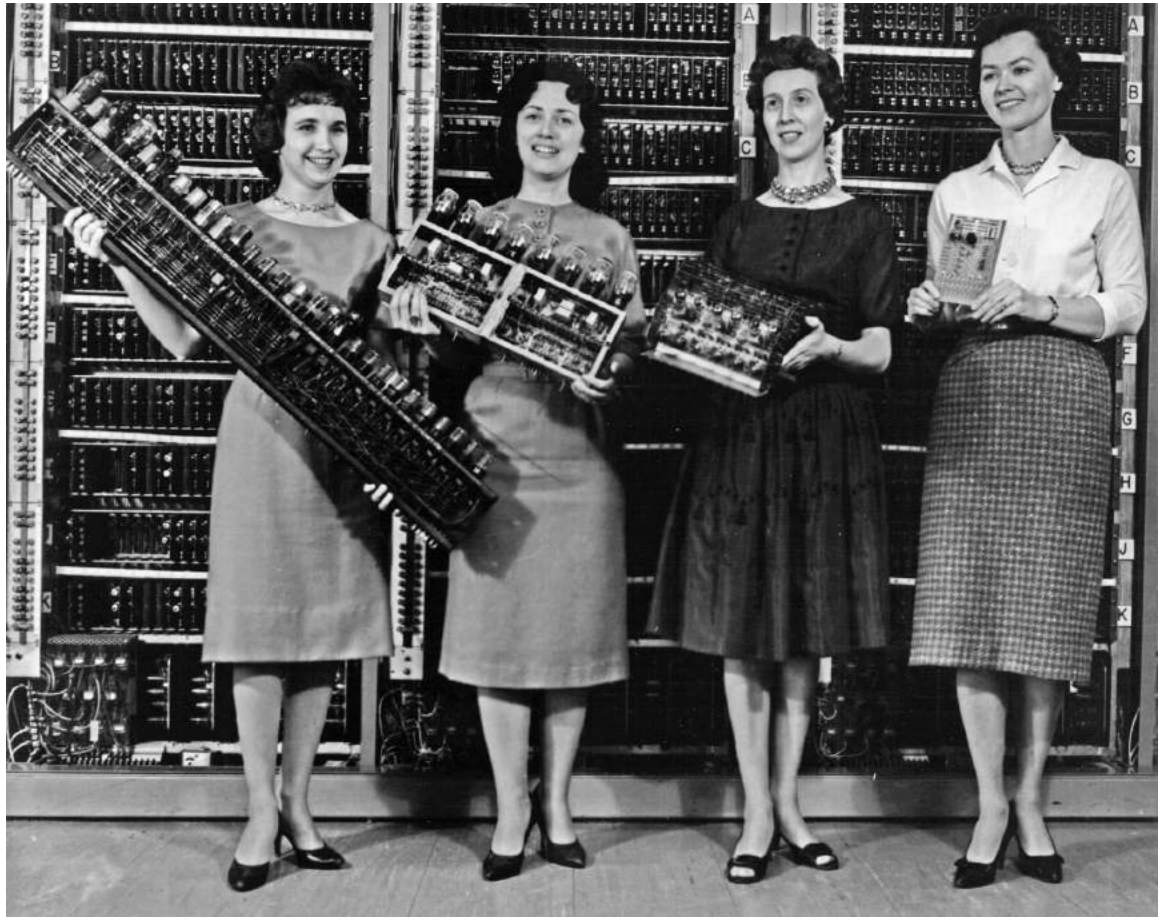
31 January 2018

CMSC421

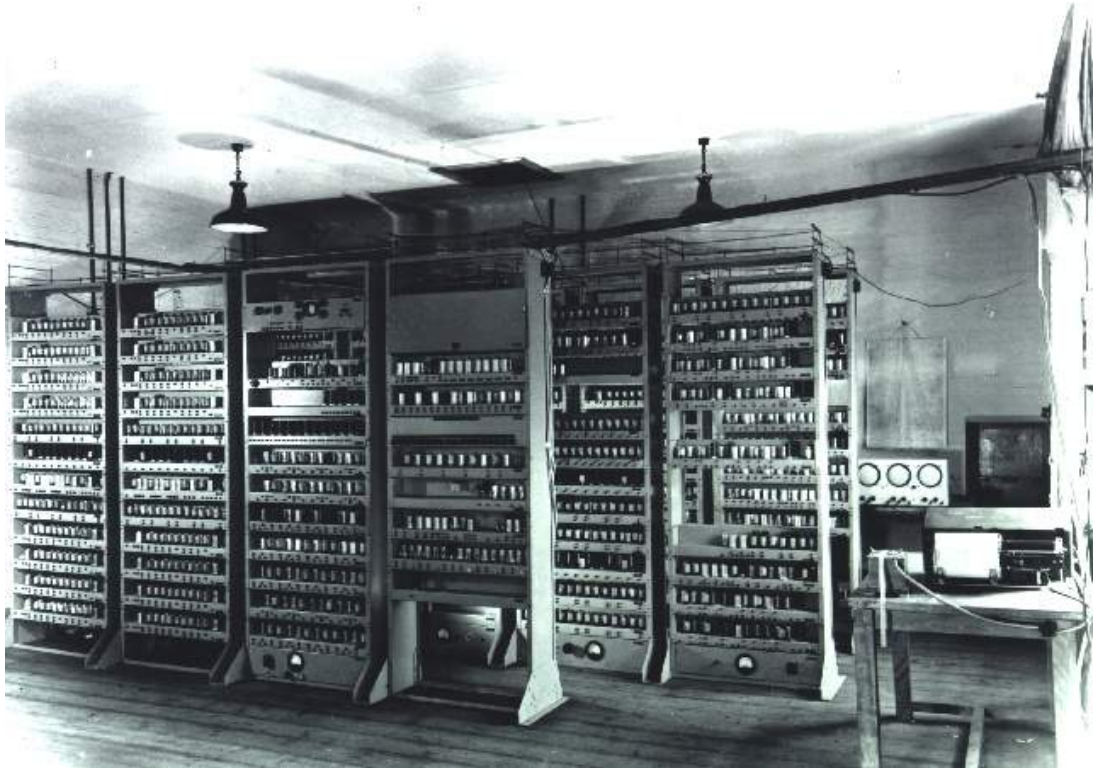
Acknowledgement

- Resources and inspiration for today:
  - https://www.grc.com/sn/sn-544.htm

Early computers



1955



Operating System Timeline

- https://en.wikipedia.org/wiki/Timeline_of_operating_systems
- 1951 EDSCA
- 1964 OS/360
- 1972 IBM VM
- 1976 Cray
- 1977 BSD
- 1978 Apple DOS
- 1984 Mac OS
- 1985 Windows 1.0

Katherine Johnson

Mainframes vs Bitcoin

Pre-Timesharing

- The computer was expensive and important to keep busy as much as possible.
- If a computer was rewinding its magnetic tape, the computer was not doing anything.
- Students would punch their card deck, take it to some guy in the machine room, and he would run it.
- You would come back later and pick up the print out.
- One or two computers per business or university.
- Expensive.

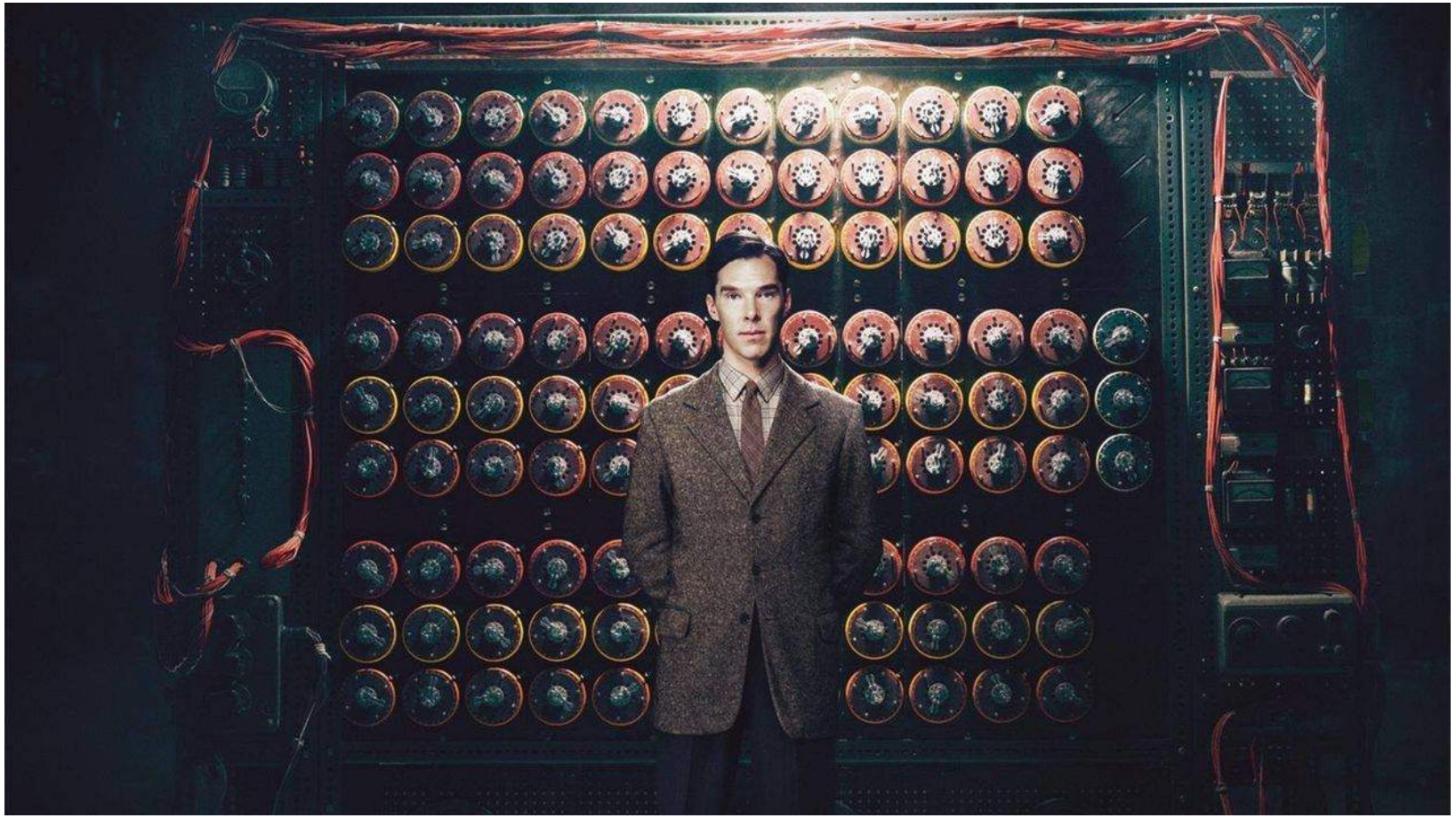
Timesharing

- Timesharing allowed users to connect to the computer and resources were shared.
- Problem, during the day, the computer was being used by whatever business was going on. Students had the lowest priority.
- Computer Science students would wake up at 0200, go over to the computer building and run their jobs. Performance was much better.
- Thus, back in the days, machines were so expensive, they were shared.
- Today, computers are ridiculously inexpensive.

Supervisor

- The original operating system was called the supervisor.
- Application programs do not have the ability to do certain things
  - Load itself
  - Buffering
  - Schedule

The Imitation Game



Scheduling

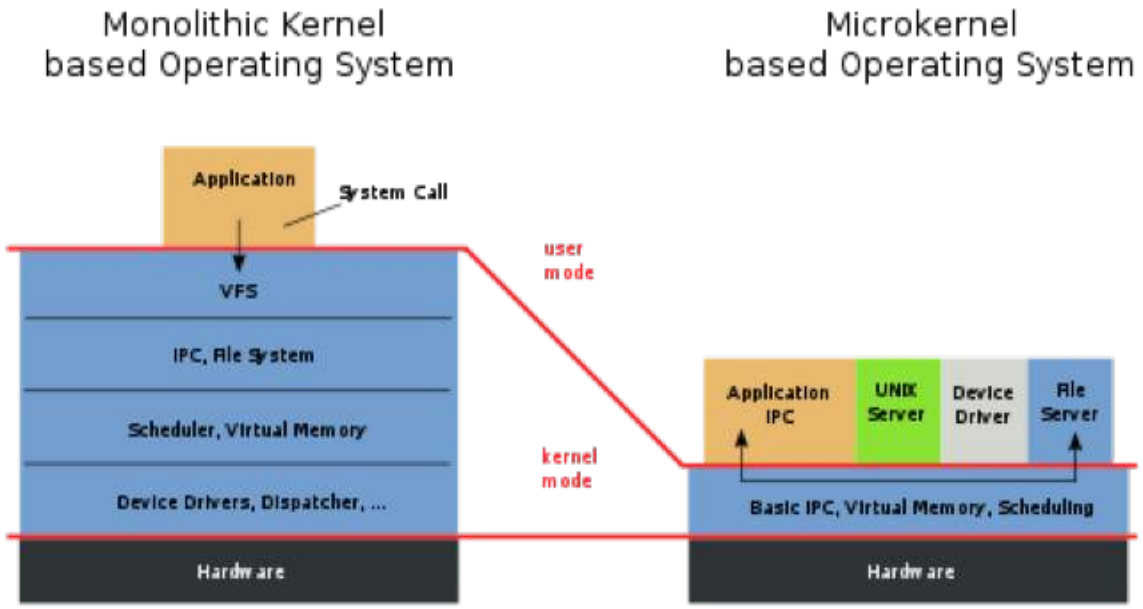
- Who has priority?
- How much memory can you have?
- Maximize the computer resources.
- Add another core?

Margaret Hamilton

https://www.youtube.com/watch?v=X1PNp_YggAA&t=192s

Micro kernel

- Minimum set of services that must be provided, that cannot be provided by the programs running or other services.
  - Loading
  - Scheduling
  - Memory management
- That is it. The ideal Micro Kernel.



Windows GDI

- Originally, the Windows OS wanted to be a microkernel. However, …
- The Microsoft Windows graphics device interface (GDI) enables applications to use graphics and formatted text on both the video display and the printer. Windows-based applications do not access the graphics hardware directly. Instead, GDI interacts with device drivers on behalf of applications.
  - https://msdn.microsoft.com/en-us/library/windows/desktop/dd145203(v=vs.85).aspx
- Microsoft decided to move the GDI into the kernel for performance.
  - Crossing back and forth from kernel to user mode was expensive.

File System

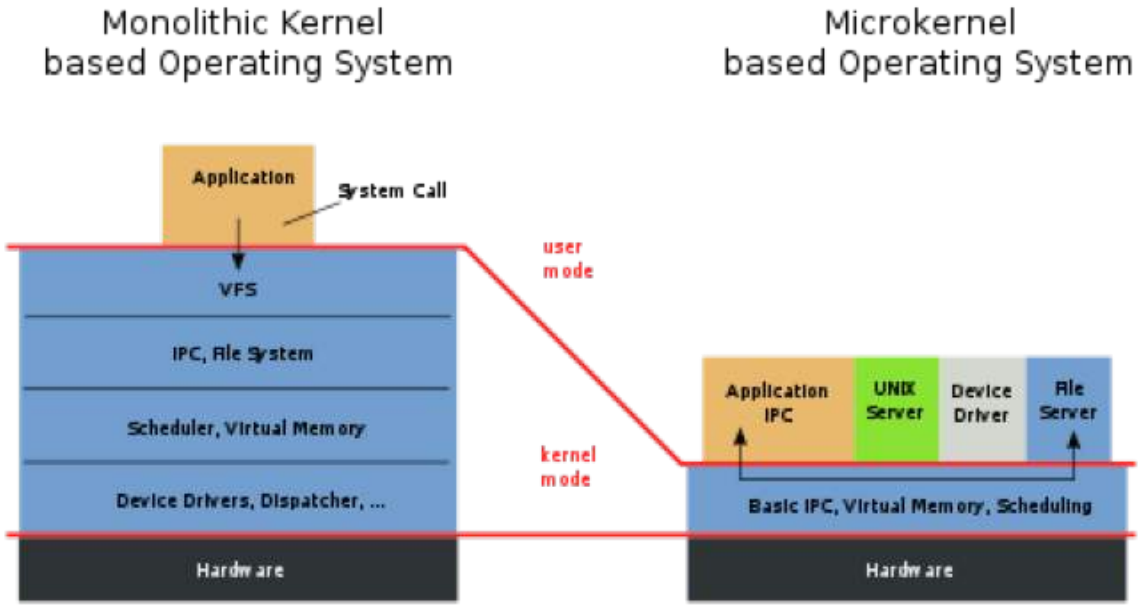
- Should this go in the kernel, or not?
- Organizes data on the disk
- An application program wants to open a file, write data, and close it.
- An application program wants to open a file, read and perform some sort of action based on the data.
- Don’t care if the file system is FAT, NTSF or ZFS.

Microkernel

- When starting with designing a kernel, everyone wants a micro kernel.
- Then for various reasons, you have crud creep.
- More stuff keeps getting put in the kernel.
- Problem. The kernel should be perfect code. No mistakes. Else, blue screen of death.
- When you keep adding code, errors will occur.
- https://www.grc.com/sn/sn-574-notes.pdf

Microkernel

- Microkernel – only necessary code is in the kernel
  - Loader – Yes
  - Scheduler – Yes
  - Memory management – Yes
  - Image rendering – No



Microkernel

- Mac OS is based on a microkernel
  - MACH
    - https://www.gnu.org/software/hurd/microkernel/mach/history.html
- Linux is definitely NOT a microkernel, as you will find out.

Last slide

- See subject.