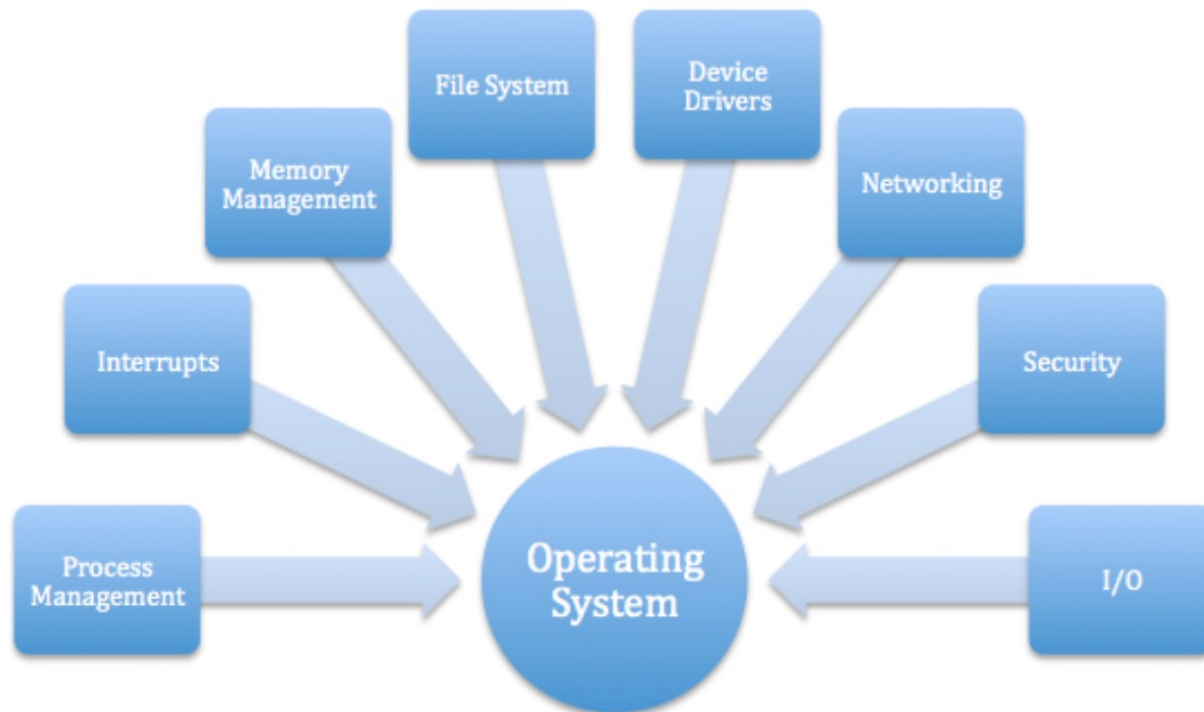# Operating Systems

Operating Systems

- Definition: *the main program in a computer that controls the way the computer works and makes it possible for other programs to function.*

- Consists of software that:
  - Controls the use of computer hardware and software resources
  - Enables user interaction via applications
  - Gives direct access to various functions outside of applications (such as copying/deleting files)
  - Updates the OS itself

Operating System Functions

- Gives applications easy but "safe" access to hardware (safe meaning it performs the desired actions without crashing the system)
- Manages sharing of data and security
- Enables use of resources, such as memory, storage, and networking

Operating Systems

- The biggest and fastest computer in the world is of no use if it cannot efficiently provide beneficial services to its users.

- Users see the computer through their application programs. These programs are ultimately executed by computer hardware.

- System software, in the form of operating systems and middleware, is the glue that holds everything together.

- The evolution of operating systems has paralleled the evolution of computer hardware.

- As hardware became more powerful, operating systems allowed people to more easily manage the power of the machine.

Operating Systems

- In the 1960s, hardware has become powerful enough to accommodate multiprogramming, the concurrent execution of more than one task.

- Multiprogramming is achieved by allocating each process a given portion of CPU time (a timeslice).

- Interactive multiprogramming systems were called timesharing systems.

- When a process is taken from the CPU and replaced by another, we say that a context switch has occurred.

Operating Layers

- Users: input data, require steps to be executed, save data or generate output

- Application: responds to requests such as saving a file by passing it along to the OS

- OS layer: ("other system software" = e.g. device drivers - software that assists the OS)

- Hardware: the physical computer. Kernel performs/controls the actions on the hardware.
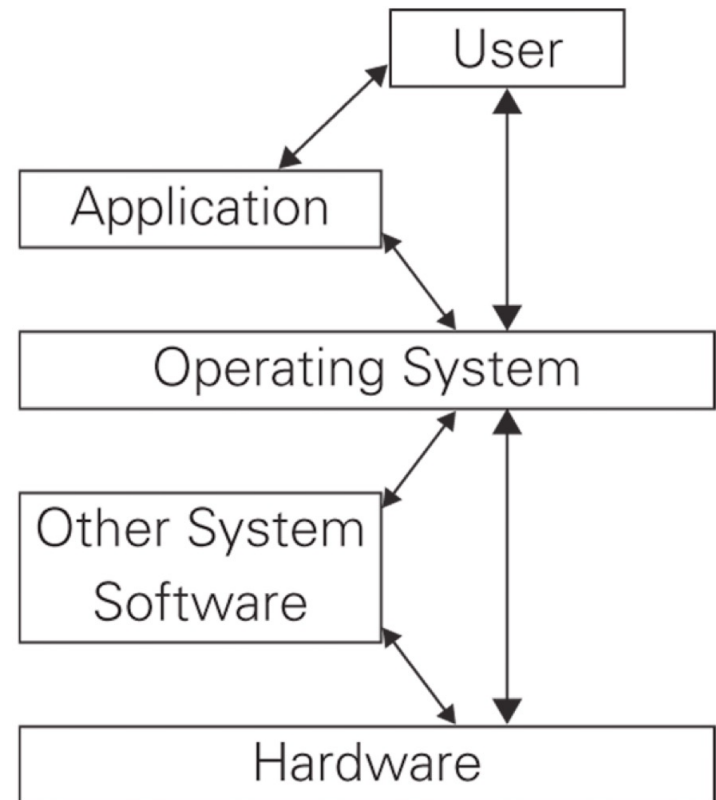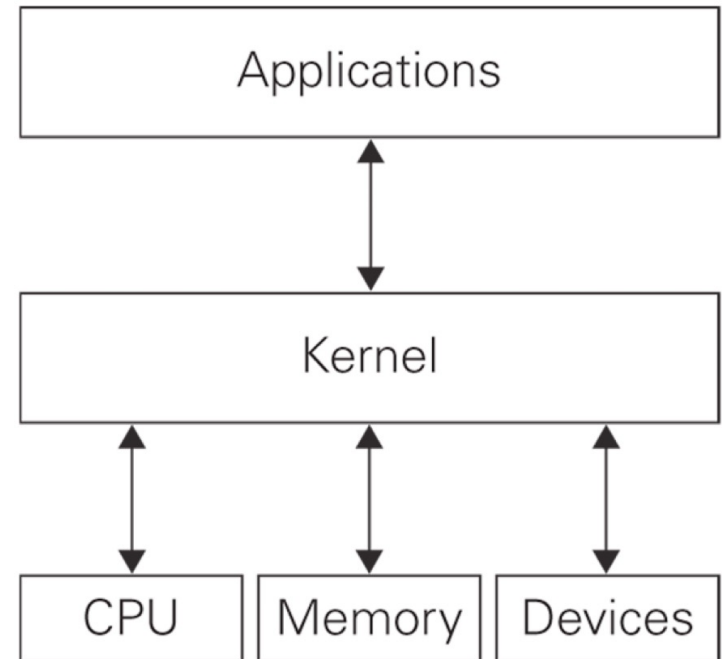
File Systems

- Different schemes of organizing and storing files on a storage device

- FAT32 (File Allocation Table)

- NTFS (New Technology File System)

- HFS+ (Hierarchical File System)

- APFS (Apple File System)

- ext4:
  - Journaling file system for Linux
  - Fourth extended filesystem
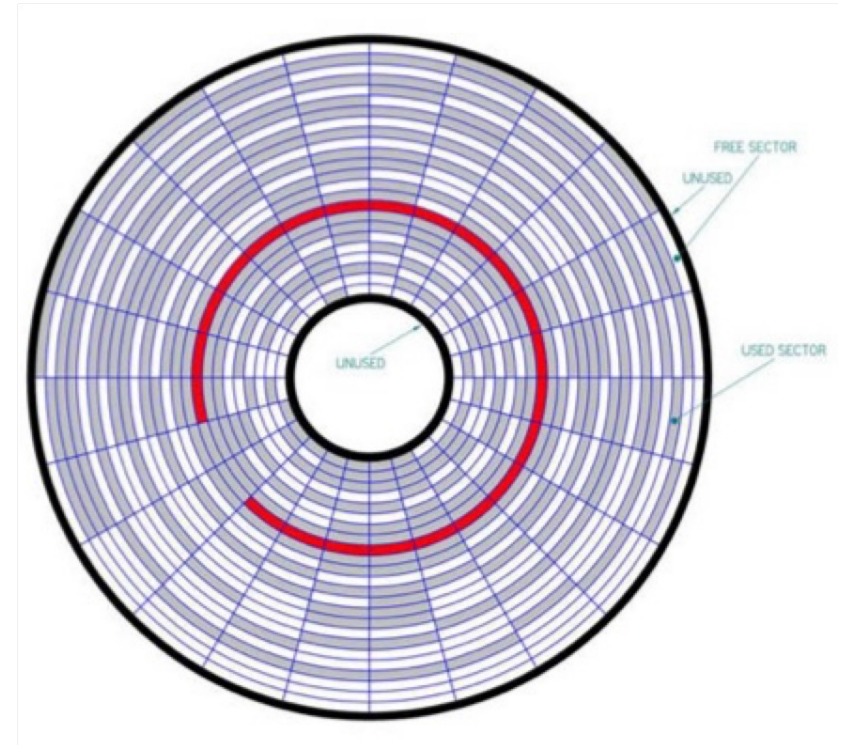
The Kernel

- Two operating system components are crucial: kernel and the system programs

- Kernel is software that oversees and exerts basic control for a computer's hardware, memory access, central processing unit (CPU), storage devices and file systems

- As the core of the operating system, the kernel performs scheduling, synchronization, memory management, interrupt handling and it provides security and protection

- Provides the necessary interface for applications to use the computer's hardware

Applications

Kernel

CPU | Memory | Devices

Command Line Interface

- Operating systems have a powerful utility that allows commands to be entered and executed on the command line

- This Command Line Interface provides the user with access to operating system functions

- All major operating systems have a command line interface:
  - Windows Command Prompt / PowerShell
  - macOS Terminal
  - Unix Terminal
  - Linux Terminal


```
pi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
pi@raspberrypi:~ $ 
```

Command Line Interface

- Operating systems have a powerful utility that allows commands to be entered and executed on the command line

- This Command Line Interface provides the user with access to operating system functions

- All major operating systems have a command line interface:
  - Windows Command Prompt / PowerShell
  - macOS Terminal
  - Unix Terminal
  - Linux Terminal

PowerShell

https://learn.microsoft.com/en-us/powershell/scripting/discover-powershell?view=powershell-7.4

Terminal Commands

Unix/Linux Command Reference — FOSSwire.com

File Commands
ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2*
if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Process Management
ps – display your currently active processes
top – display all running processes
kill *pid* – kill process id *pid*
killall *proc* – kill all processes named *proc* *
bg – lists stopped or background jobs; resume a stopped job in the background
fg – brings the most recent job to foreground
fg *n* – brings job *n* to the foreground

File Permissions
chmod *octal file* – change the permissions of *file* to *octal*, which can be found separately for user, group, and world by adding:
- 4 – read (r)
- 2 – write (w)
- 1 – execute (x)

Examples:
chmod 777 – read, write, execute for all
chmod 755 – rwx for owner, rx for group and world
For more options, see **man chmod**.

SSH
ssh *user@host* – connect to *host* as *user*
ssh -p *port user@host* – connect to *host* on port *port* as *user*
ssh-copy-id *user@host* – add your key to *host* for *user* to enable a keyed or passwordless login

Searching
grep *pattern files* – search for *pattern* in *files*
grep -r *pattern dir* – search recursively for *pattern* in *dir*
command | grep *pattern* – search for *pattern* in the output of *command*
locate *file* – find all instances of *file*

System Info
date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression
tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*
gzip -d *file.gz* – decompresses *file.gz* back to *file*

Network
ping *host* – ping *host* and output results
whois *domain* – get whois information for *domain*
dig *domain* – get DNS information for *domain*
dig -x *host* – reverse lookup *host*
wget *file* – download *file*
wget -c *file* – continue a stopped download

Installation
Install from source:
./configure
make
make install
dpkg -i *pkg.deb* – install a package (Debian)
rpm -Uvh *pkg.rpm* – install a package (RPM)

Shortcuts
Ctrl+C – halts the current command
Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background
Ctrl+D – log out of current session, similar to **exit**
Ctrl+W – erases one word in the current line
Ctrl+U – erases the whole line
Ctrl+R – type to bring up a recent command
!! – repeats the last command
exit – log out of current session

* use with extreme caution.

```
kali@kali:~$ sudo su
[sudo] password for kali:
root@kali:/home/kali# useradd SomeUser
root@kali:/home/kali# chown SomeUser:SomeUser ./Devin/myFile
root@kali:/home/kali# ls -l ./Devin/
total 4
-rw-r--r-- 1 SomeUser SomeUser 10 Feb  5 01:13 myFile
root@kali:/home/kali# chmod 770 ./Devin/myFile
root@kali:/home/kali# ls -l ./De
Desktop/ Devin/
root@kali:/home/kali# ls -l ./Devin/
total 4
-rwxrwx--- 1 SomeUser SomeUser 10 Feb  5 01:13 myFile
root@kali:/home/kali#
```