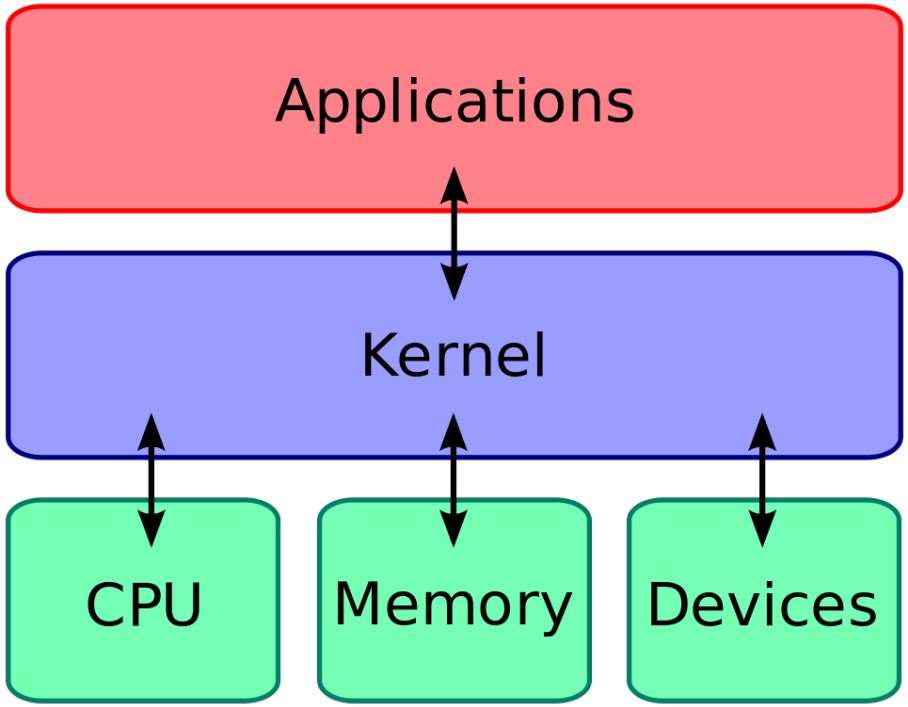
COMP20200 Unix Programming

Lecture 3

CS, University College Dublin, Ireland

What is the Kernel?

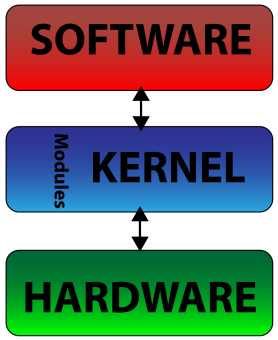
- Fundamental part of operating system
- Connects software to physical devices
- Provides low-level abstraction layer
- Applications make requests through system calls



Monolithic vs Microkernels

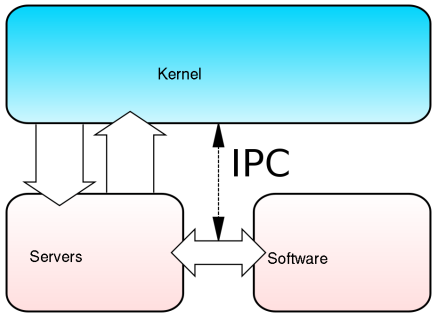
Monolithic Kernel

- All OS instructions in same address space
- Powerful hardware access
- Dependencies between system components
- Easier to implement
- Harder to maintain



Microkernel

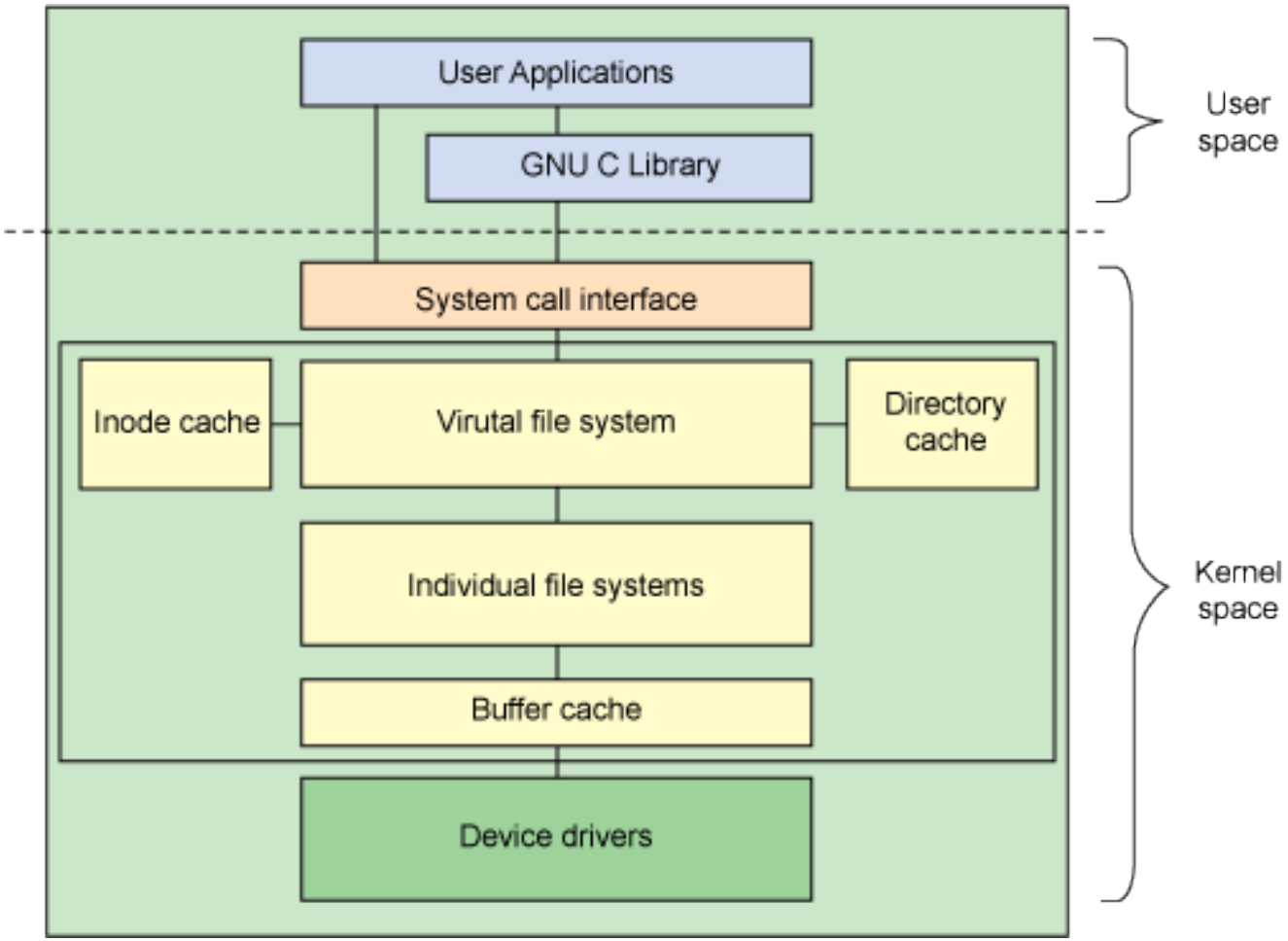
- Functionality moved to "servers"
- Servers communicate through a "minimal" kernel with Inter-Process Communication (IPC)
- Harder to implement
- Easier to maintain



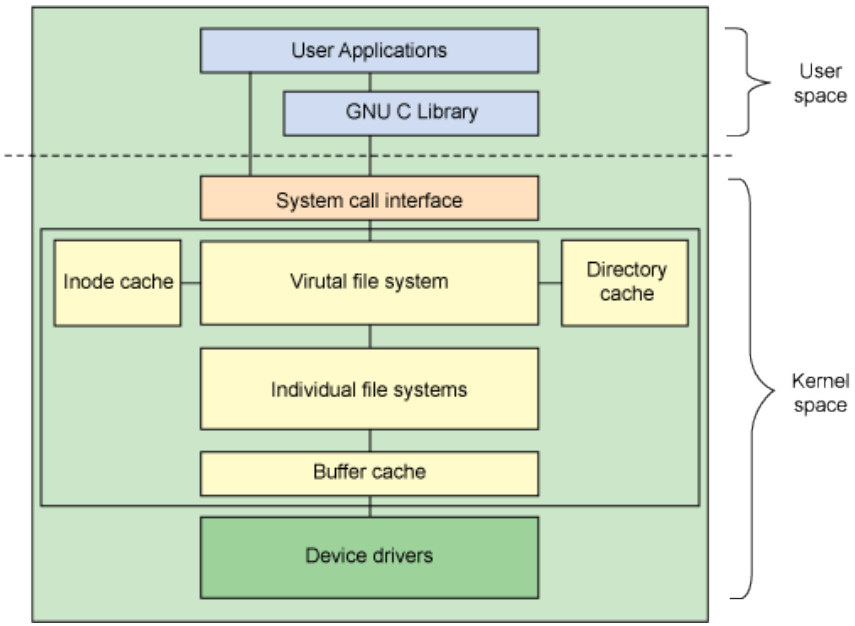
Further reading: `http://en.wikipedia.org/wiki/Kernel_(computing)`

See also: Hybrid and Exokernels

Virtualization of Memory and File systems

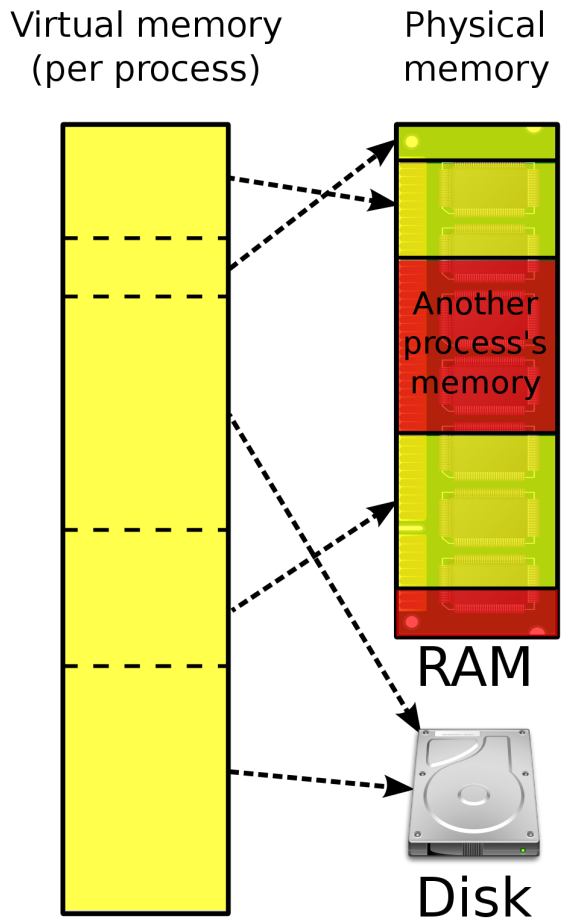


- System programmers don't program hardware directly.
- Device drivers and kernel provide a virtual memory and file systems.
- OS segregates virtual memory into kernel space and user space
- "Userland" applications run in user space.
- Interface through system calls.
- System programming often involves memory management of this virtual memory.



Virtual Memory

- Physical memory is fragmented.
- Unix provides each process with a continuous block of virtual memory.
- Every application has unique address space.
- Equal in size to architecture of system.
  - 32-bit: each process can address 4 GB of memory
  - 64-bit: in theory can address 16 exabytes (16×10^6 TB)
    - in practice its less and architecture dependent.
  - Top section of address space is reserved for kernel.

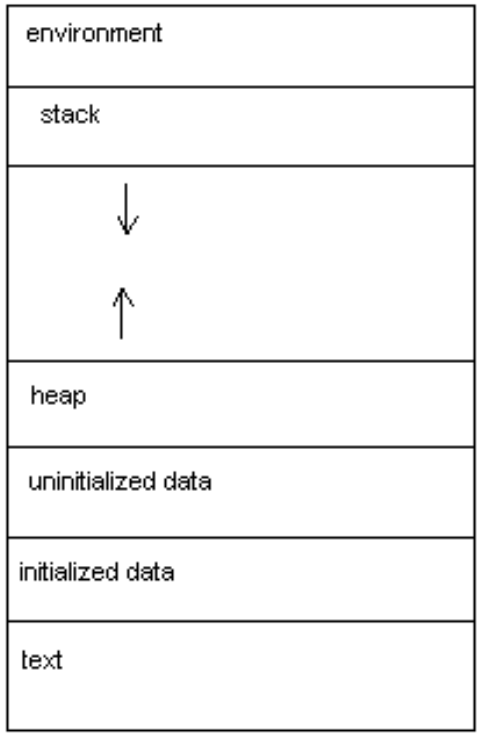


- If a process wants to access more than available RAM, hard disk can be mapped (*swap*), (process is not aware).

Virtual Memory

A process's address space typically has 6 sections:

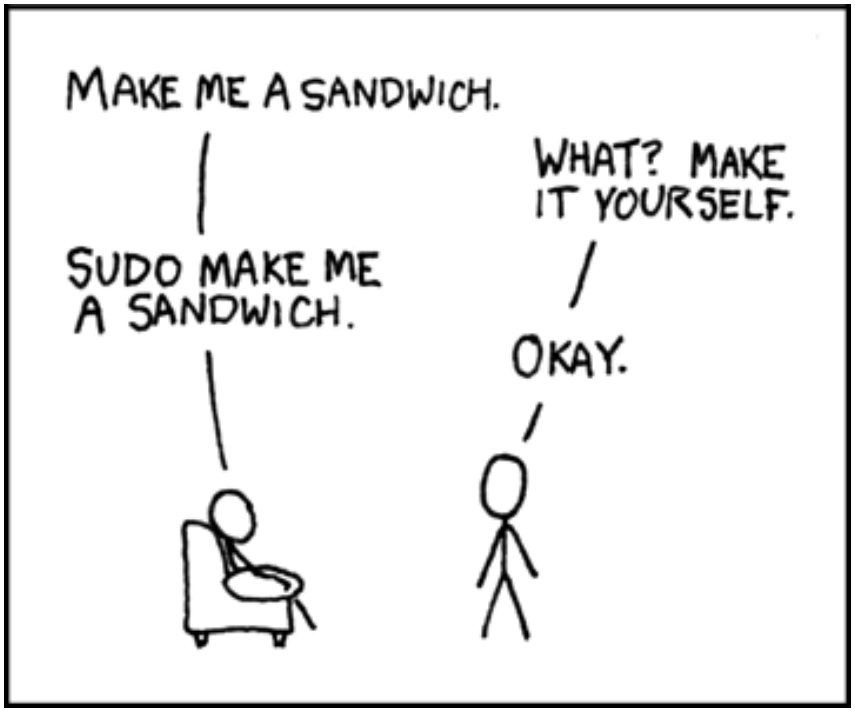
- Environment
  - Environment variables
  - command line arguments
- Stack
  - Function arguments
  - Return values
  - Automatic variables
- Heap
  - Dynamic allocation
- Data (uninitialized/initialized)
  - Static & global variables
- Text
  - The program code



Virtual memory organization

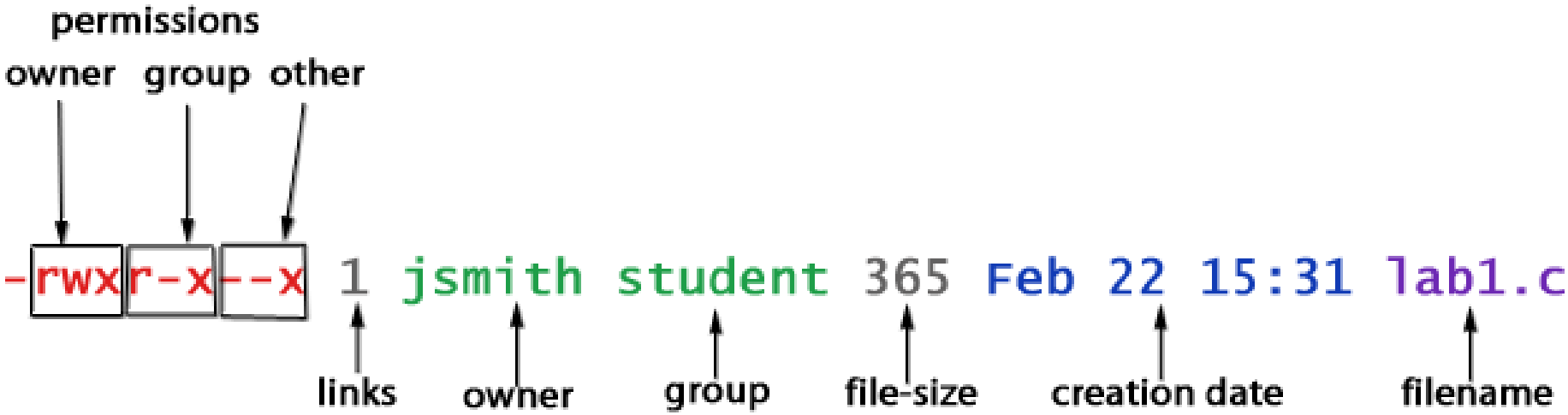
Superuser

- Root account for administration.
- A user in user space like other user accounts
- `sudo` is a program to allow users to run programs with security privileges of another user (usually root).

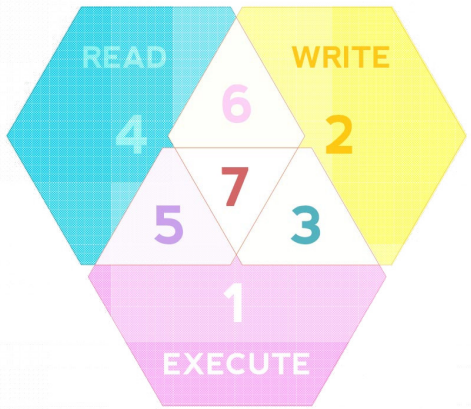


File Permissions

Result of `$ ls -l` :



- Modify with
  - `chmod`
  - `chown`
  - `chgrp`



Finding files

- `whereis` locate the binary, source, and manual page files for a command
- `which` locate a command
- `locate` find files by name
- `find` search for files in a directory hierarchy
  - Find is an extremely powerful tool. For example to find all .jpgs that are greater then 5MB: `find . -size +5M -name "*.jpg" -exec ls -l {} \;`

Common commands (a non-exhaustive list)

```
man
ls
cd
pwd
history
mv
cp
rm *
cat
touch
mkdir
head
tail
more
less
diff
```

```
which
gcc
wc
echo
chmod *
chown *
chgrp
grep
find
locate
sort
ln
tar
du
df
```

```
w
who
whoami
finger
passwd *
top
ps
kill *
pgrep
pkill *
last
ssh
rsync
scp
date
time
```

* be careful running this command

- Command line tip(s) of the day:
  - `history` command lists previous commands entered.
  - Ctrl+r allows searching (and then executing) of this history.
- Vi tip(s) of the day:
  - `y` Yanks text (copy)
  - `p` Puts text (paste)
  - `d` deletes text (cut)
  - `yy` yanks whole line
  - `dd` deletes whole line
  - `v` Enters visual mode, for highlighting text for deleting or yanking