

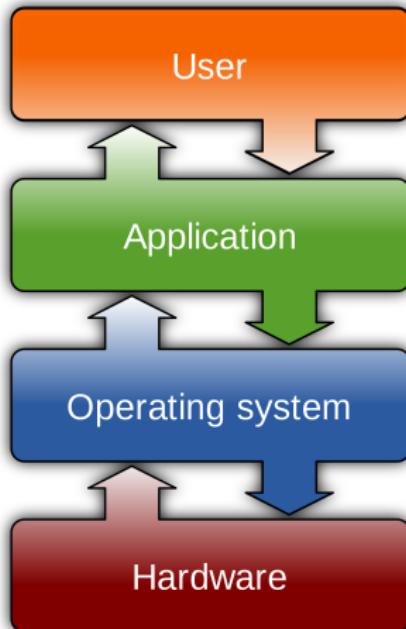
Operating Systems

- key concepts & security principles -

Myrto Arapinis
School of Informatics
University of Edinburgh

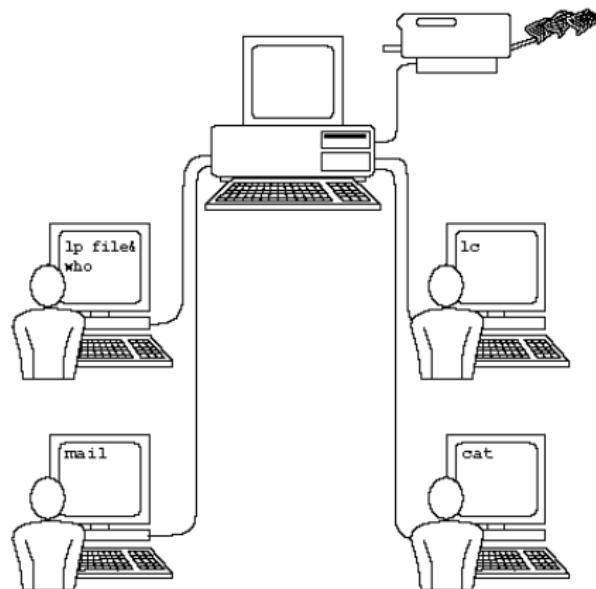
Operating systems

- ▶ An OS provides the interface between the users of a computer and that computer's hardware.
- ▶ The OS handles the management of low-level hardware resources:
 - disk drives,
 - CPU,
 - RAM,
 - I/O devices, and
 - network interfaces.



Multi-users

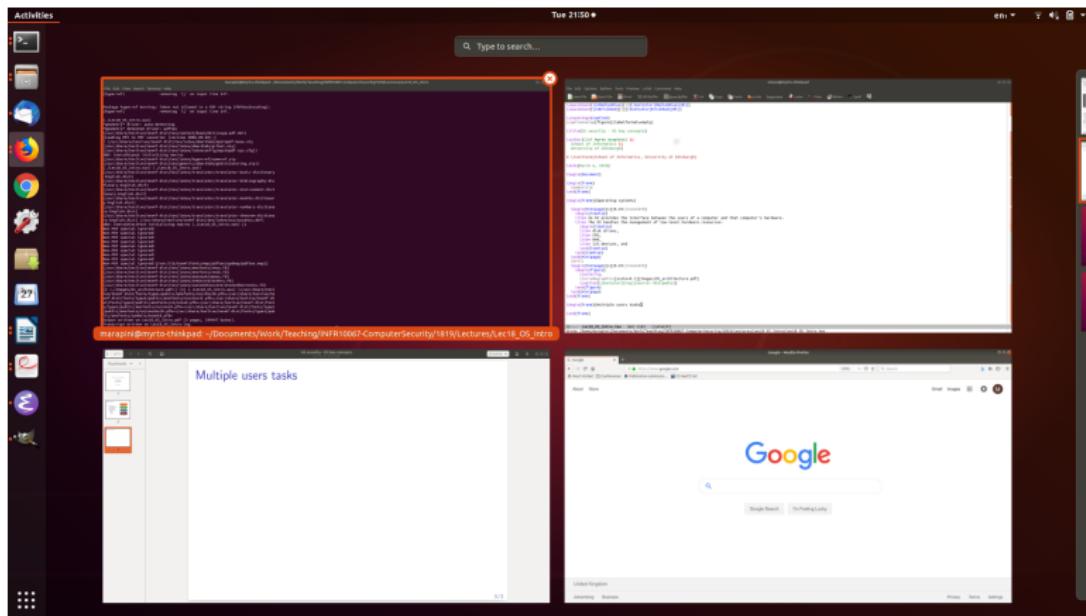
OSes must allow for multiple users with potentially different levels of access to the same computer.



The OS needs to have in place mechanisms to **isolate different users**.

Multi-tasking

OSes must allow multiple application programs to run at the same time



The OS needs to have in place mechanisms to **isolate different applications running**.

Essential Unix architecture

Kernel

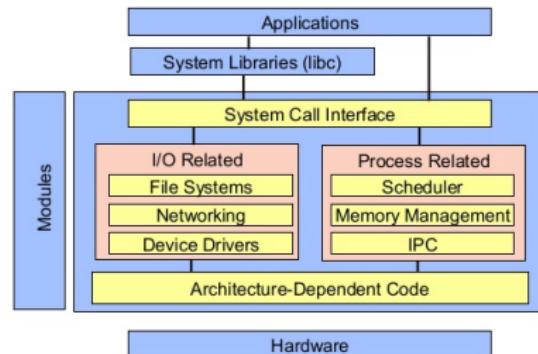
- key component of the OS
- supports secure sharing of low-level resources between users/applications
- kernel limits how applications access computer resources

Execution modes

- User mode - access to resources through syscall to kernel
- Kernel mode - direct access to resources

System calls are usually contained in a collection of programs, eg. a library such as the C library `libc`:

- `open()` `close()` `read()` `write()`
- `wait()` `fork()` `exec()` `exit()`

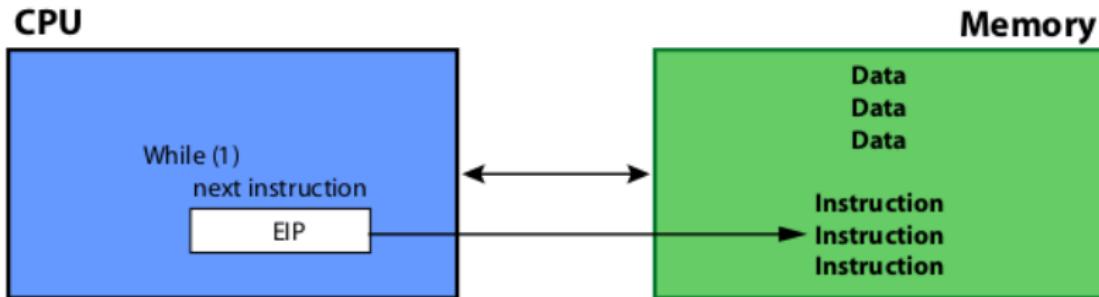


Processes and process management

- ▶ A process is an instance of a program that is currently executing
- ▶ To actually be executed the program must be loaded into RAM and uniquely identified
- ▶ Each process running is identified by a unique process ID (pid)
- ▶ To a pid, we can associate its CPU time, memory usage, user ID (uid), program name, etc
- ▶ A process might control other processes (fork)
- ▶ Child process inherits context from parent process

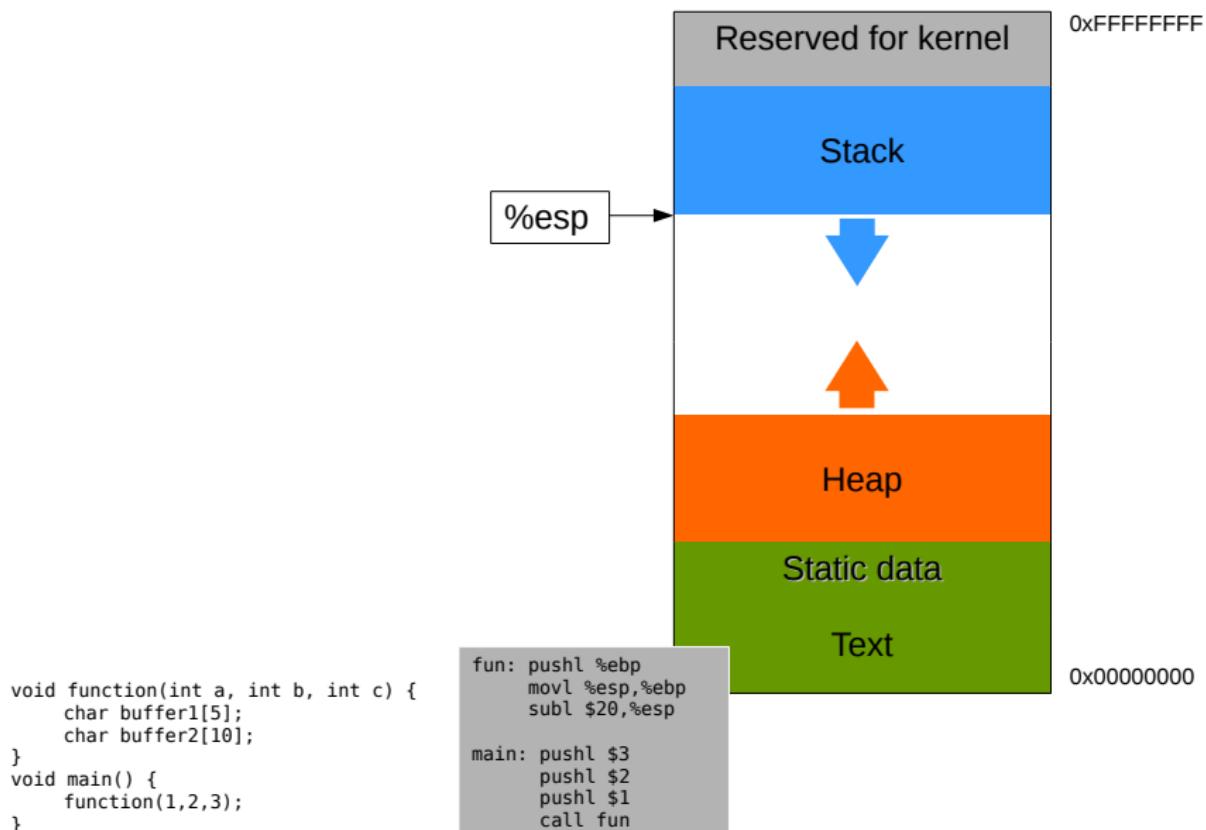
```
marapini@myrto-thinkpad: ~/Documents/Work/Teaching/INFR10067-ComputerSecurity/1819/Lectures/Lec18_05_Intro$ ps -ef
File Edit View Search Terminal Help
marapini@myrto-thinkpad: ~/Documents/Work/Teaching/INFR10067-ComputerSecurity/1819/Lectures/Lec18_05_Intro$ ps -ef
UID      PID  PPID  C STIME TTY      TIME CMD
root      1  0 Mar03 ?    00:00:29 /sbin/init splash
root      2  0 Mar03 ?    00:00:00 [kthread]
root      4  2 0 Mar03 ?    00:00:00 [migration/0]
root      6  2 0 Mar03 ?    00:00:00 [ren_percpu_wq]
root      7  2 0 Mar03 ?    00:00:01 [ksoftirqd/0]
root      8  2 0 Mar03 ?    00:00:14 [rcu_sched]
root      9  2 0 Mar03 ?    00:00:00 [rcu_bh]
root     10  2 0 Mar03 ?    00:00:00 [migration/0]
root     11  2 0 Mar03 ?    00:00:00 [watchdog/0]
root     12  2 0 Mar03 ?    00:00:00 [cpuhp/0]
root     13  2 0 Mar03 ?    00:00:00 [cpuhp/1]
root     14  2 0 Mar03 ?    00:00:00 [watchdog/1]
root     15  2 0 Mar03 ?    00:00:00 [migration/1]
root     16  2 0 Mar03 ?    00:00:00 [ksoftirqd/1]
root     18  2 0 Mar03 ?    00:00:00 [ren_percpu_wq]
root     19  2 0 Mar03 ?    00:00:00 [cpuhp/2]
root     20  2 0 Mar03 ?    00:00:00 [watchdog/2]
root     21  2 0 Mar03 ?    00:00:00 [migration/2]
root     22  2 0 Mar03 ?    00:00:01 [ksoftirqd/2]
root     24  2 0 Mar03 ?    00:00:00 [migrate/2:0]
root     25  2 0 Mar03 ?    00:00:00 [cpuhp/3]
root     26  2 0 Mar03 ?    00:00:00 [watchdog/3]
root     27  2 0 Mar03 ?    00:00:00 [migration/3]
root     28  2 0 Mar03 ?    00:00:00 [ksoftirqd/3]
root     30  2 0 Mar03 ?    00:00:00 [kworker/j:0:0]
root     31  2 0 Mar03 ?    00:00:00 [kdevtmpfs]
root     32  2 0 Mar03 ?    00:00:00 [ksoftirqd/4]
root     33  2 0 Mar03 ?    00:00:00 [rcu_tasks_kthre]
root     34  2 0 Mar03 ?    00:00:00 [kauditd]
root     38  2 0 Mar03 ?    00:00:00 [khungtaskd]
root     39  2 0 Mar03 ?    00:00:00 [oom_reaper]
root     40  2 0 Mar03 ?    00:00:00 [writeback]
root     42  2 0 Mar03 ?    00:00:00 [kblockd]
root     42  2 0 Mar03 ?    00:00:00 [kwd]
root     43  2 0 Mar03 ?    00:00:00 [khugepaged]
root     44  2 0 Mar03 ?    00:00:00 [crypto]
root     45  2 0 Mar03 ?    00:00:00 [kintegrityd]
root     46  2 0 Mar03 ?    00:00:00 [kblockd]
root     48  2 0 Mar03 ?    00:00:00 [ksoftirqd/0]
root     49  2 0 Mar03 ?    00:00:00 [rd]
root     50  2 0 Mar03 ?    00:00:00 [edac_mc]
root     51  2 0 Mar03 ?    00:00:00 [devfreq_wq]
root     52  2 0 Mar03 ?    00:00:00 [watchdogd]
root     53  2 0 Mar03 ?    00:00:00 [kswapd0]
root     56  2 0 Mar03 ?    00:00:00 [kmemleak]
root     98  2 0 Mar03 ?    00:00:00 [kthread]
root     99  2 0 Mar03 ?    00:00:00 [acpi_thermal_throttler]
root    103  2 0 Mar03 ?    00:00:00 [ip6v6_addrconf]
root   112  2 0 Mar03 ?    00:00:00 [kstrp]
root   129  2 0 Mar03 ?    00:00:00 [charger_manager]
root   376  2 0 Mar03 ?    00:00:00 [ksoftirqd/1]
root   180  2 0 Mar03 ?    00:00:10 [t915/signal:0]
root   181  2 0 Mar03 ?    00:00:00 [t915/signal:1]
```

x86 CPU/Memory

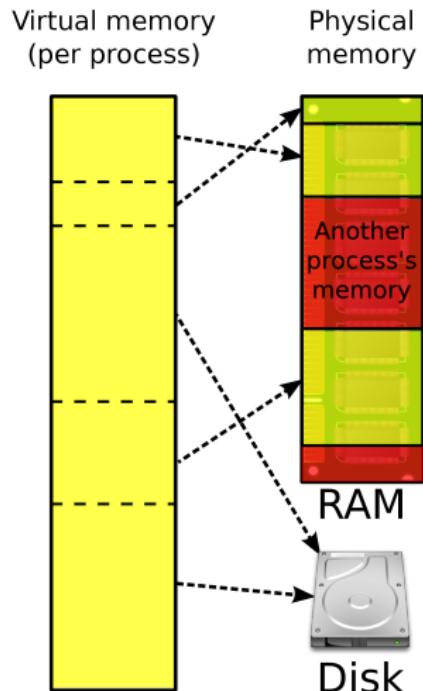


- To actually be executed the program must be loaded into RAM and uniquely identified
- The RAM memory allocated to a process is its address space
- It contains both the code for the running program, its input data, and its working memory
- CPU interprets instructions - %eip points to next instruction

x86 process memory layout (simplified)



Virtual memory



- ▶ Common technique used in a computer's OS
- ▶ Sometimes available RAM is not enough to run several programs at one time. This is where virtual memory comes in.
- ▶ A system using virtual memory uses a section of the hard drive to emulate RAM - secondary memory treated as though it were main memory
- ▶ A memory management unit (MMU) maps a logical address space to a corresponding physical address.

Live CD attacks on memory

The attack:

1. Attacker with physical access to computer powers off the computer (without properly shutting down)
2. Attacker boots to different OS via external media
3. Attacker retrieves the Pagefile.sys, Swapfile.sys, Hiberfile.sys files
4. Attacker gains access to passwords and sensitive information that were stored in memory

Live CD attacks on memory

The attack:

1. Attacker with physical access to computer powers off the computer (without properly shutting down)
2. Attacker boots to different OS via external media
3. Attacker retrieves the Pagefile.sys, Swapfile.sys, Hiberfile.sys files
4. Attacker gains access to passwords and sensitive information that were stored in memory

Mitigation:

Hard disk encryption must be used !

Security principles

Defence-in-depth



- ▶ Security protections built in multiple layers of the system: if one mechanism fails, another steps up immediately behind to thwart attacks
- ▶ Firewalls, intrusion detection and protection systems, network segmentation, anti-virus, least privilege, strong passwords, patch management

Least privilege



- ▶ Users and programs should only access the data and resources required to perform its function

Least privilege



Brightest Flashlight Free ®
Version 2.4.2 can access

- Location
 - approximate location (network-based)
 - precise location (GPS and network-based)
- Photos/Media/Files
 - read the contents of your USB storage
 - modify or delete the contents of your USB storage
- Camera/Microphone
 - take pictures and videos
- Wi-Fi connection information
 - view Wi-Fi connections
- Device ID & call information
 - read phone status and identity

Updates to Brightest Flashlight Free ® may automatically add additional capabilities within each group. [Learn more](#)

- ▶ Users and programs should only access the data and resources required to perform its function
- ▶ A torch application does not need access to your location, photos, camera, microphone, wifi, device id, to perform its intended task!

Privilege separation

App permissions		SMS permissions	
Calendar	4 of 8 apps allowed	Caping	OFF
Camera	4 of 16 apps allowed	Contacts	OFF
Contacts	7 of 22 apps allowed	Duo	OFF
Location	2 of 17 apps allowed	Google	ON
Microphone	5 of 14 apps allowed	Google Play services	ON
Phone	7 of 20 apps allowed	Google Play Store	ON
SMS	5 of 13 apps allowed	Kaspersky Internet Security	OFF
Storage		Maps	OFF
		Messages	ON
		Messenger	OFF

- ▶ Segment the system into components to which we can limit access
- ▶ Will limit the damage caused by a security break of any individual component

Open design



- ▶ The security of a mechanism should not depend on its secrecy
- ▶ The design and implementation details always get leaked (!)

Economy of mechanism



- ▶ When designing a security mechanism keep it simple!
- ▶ It will facilitate the job of security researchers and allow verification
- ▶ It will facilitate the task of developers and avoid bugs
- ▶ It will facilitate the life of users and avoid misuses

More security principles

Fail-safe defaults - default configuration should be conservative, eg. new user should be granted least privileges by default

Complete mediation – every access to a resource must be checked for compliance with security policy

Usable security – UIs and security mechanisms should be designed with the ordinary user in mind – the users should be supported in interacting in a secure way with the system – you can't blame users !

...

What we learned today

Many tasks handled by the OS relate to fundamental security problems.

1. OS concepts

- basic tasks of the OS
- security concerns arise from multiple users and multiple processes
- processes and process management
- x86 runtime memory

2. Security design principles

- Defence-in-depth
- Least privilege
- Privilege separation
- Open design
- Economy of mechanism