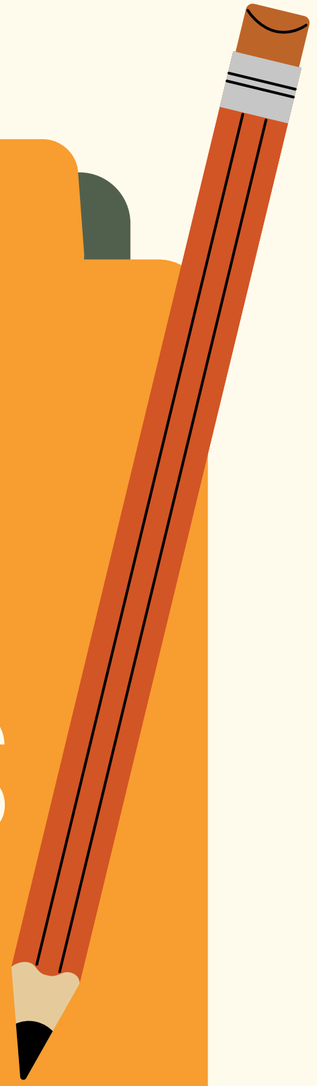
Operating System

PROCESS & THREADS

Group 1 Section 02

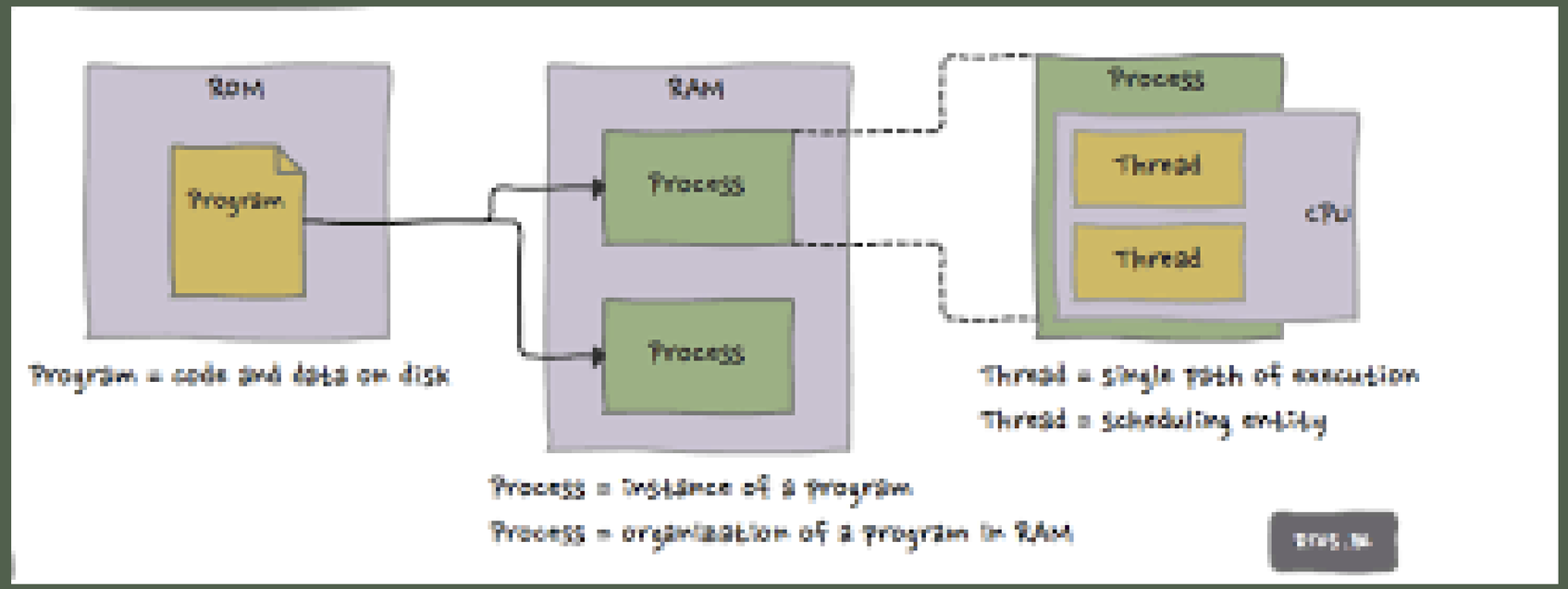


BIG QUESTION!

How does a device could multitask work? Even when we only open one app at that time, still some other background activities such as notifications are running. How exactly all these things works?

Program, process and threads diagram

Program, process and threads



Remember: ROM & RAM is a memory

Program

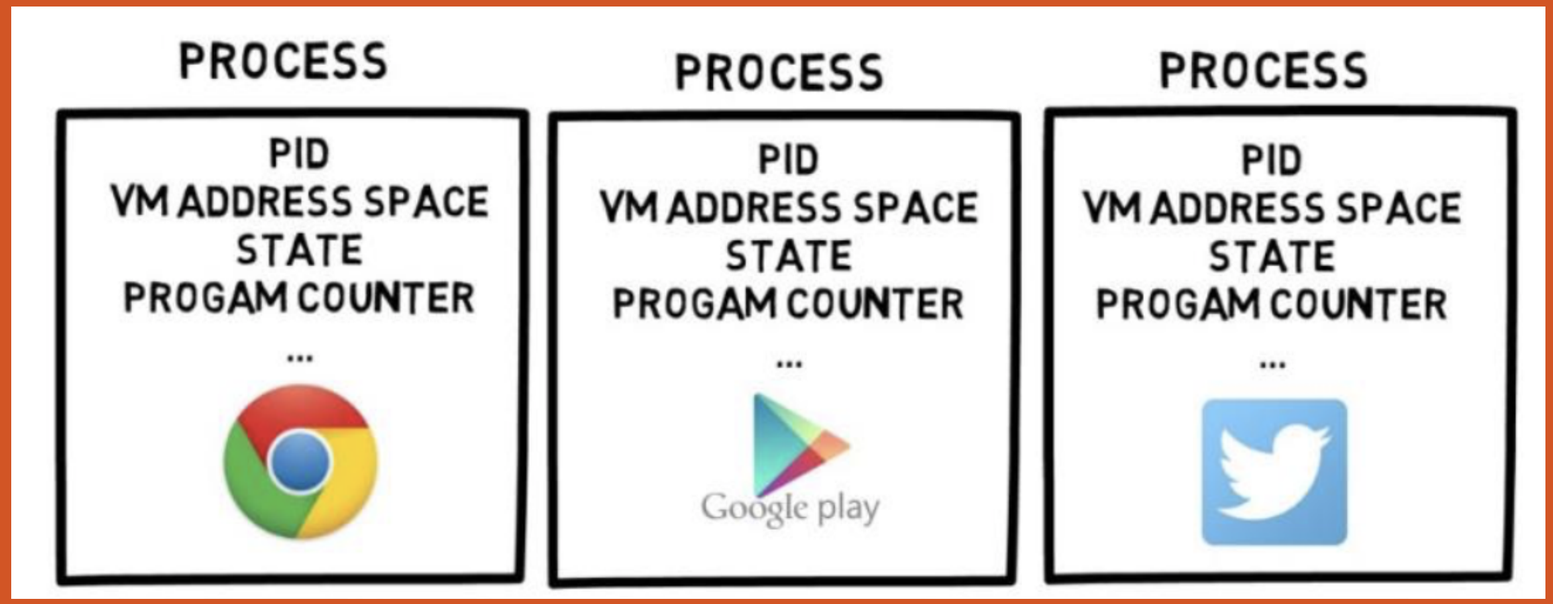
1	The code to process tasks on a computer.
2	Types of programs: a) programs for computer functionality b) programs part of operating system c) programs to fulfill particular tasks (applications, word processing, web browsing, emailing, etc)
3	Stored in memory for computer to execute.
4	End results: text file of code that compiled into binary form (CPU only understand instructions in binary form)

Process

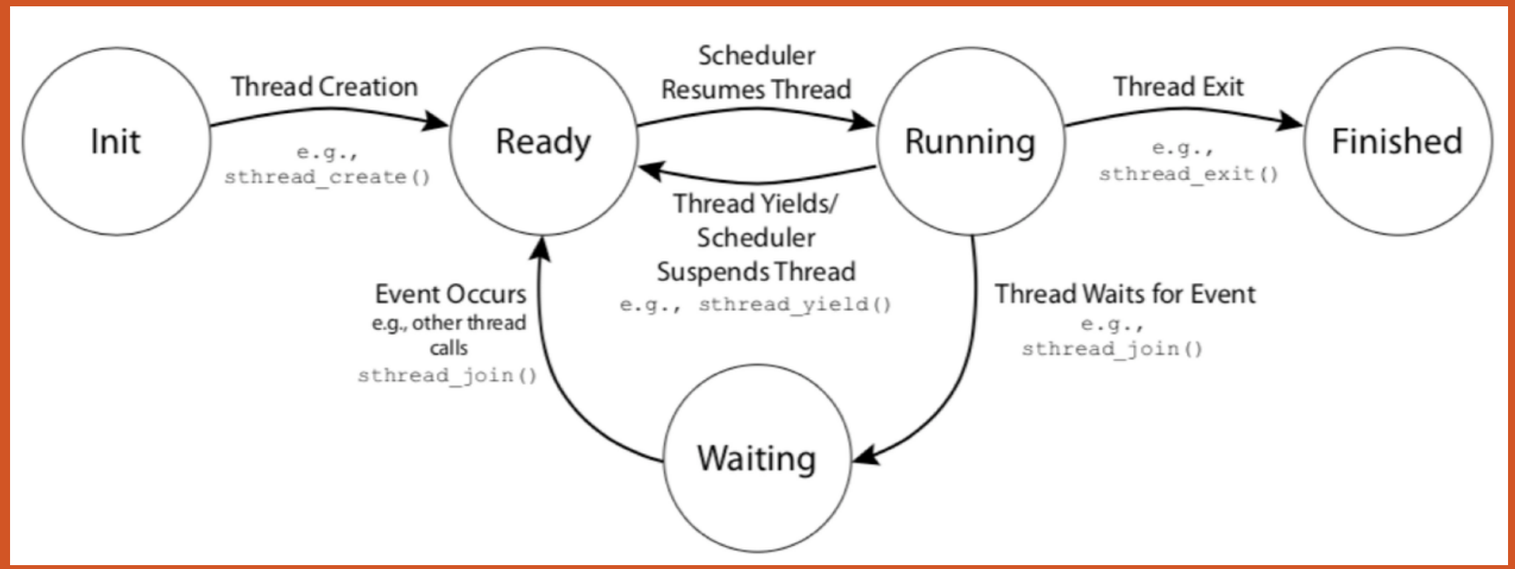
1	A program that has been loaded into memory along with all resources needed to operates.
2	Operating system act as a brain to allocate and manage all the resources which turn a programs into a running process. Example of OS: masOS, Microsoft Windows, Linux, Android, etc.
3	A single running program have multiple instances, also known as a processes.
4	Each process have its own unique ID (PID/Process ID) to identify their memory address space. It stored some state information (running, sleeping, stopped and zombie) in its own memory. Thus, it running privately from other processes.

Process

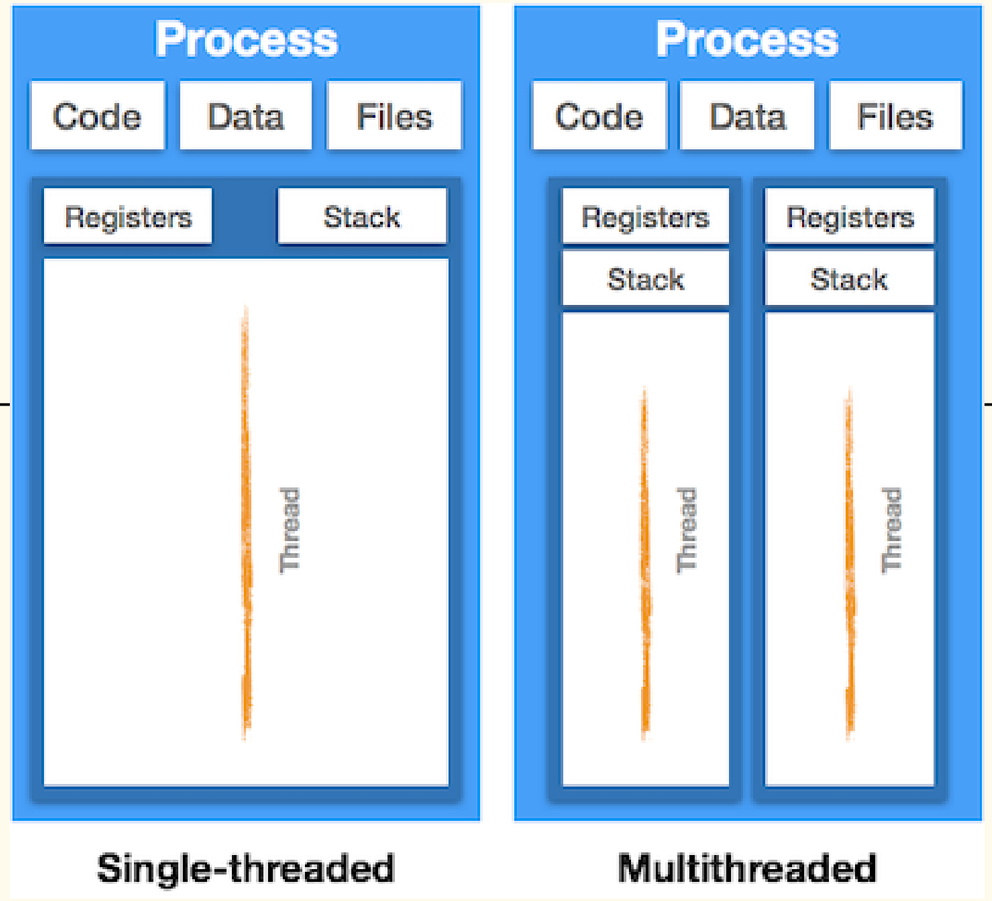
5	Switching one process to another require some time for saving and loading register, memory maps, and other resources.
6	Benefits: when one process corrupt, another process can still run. (eg: quit a freezes applications)



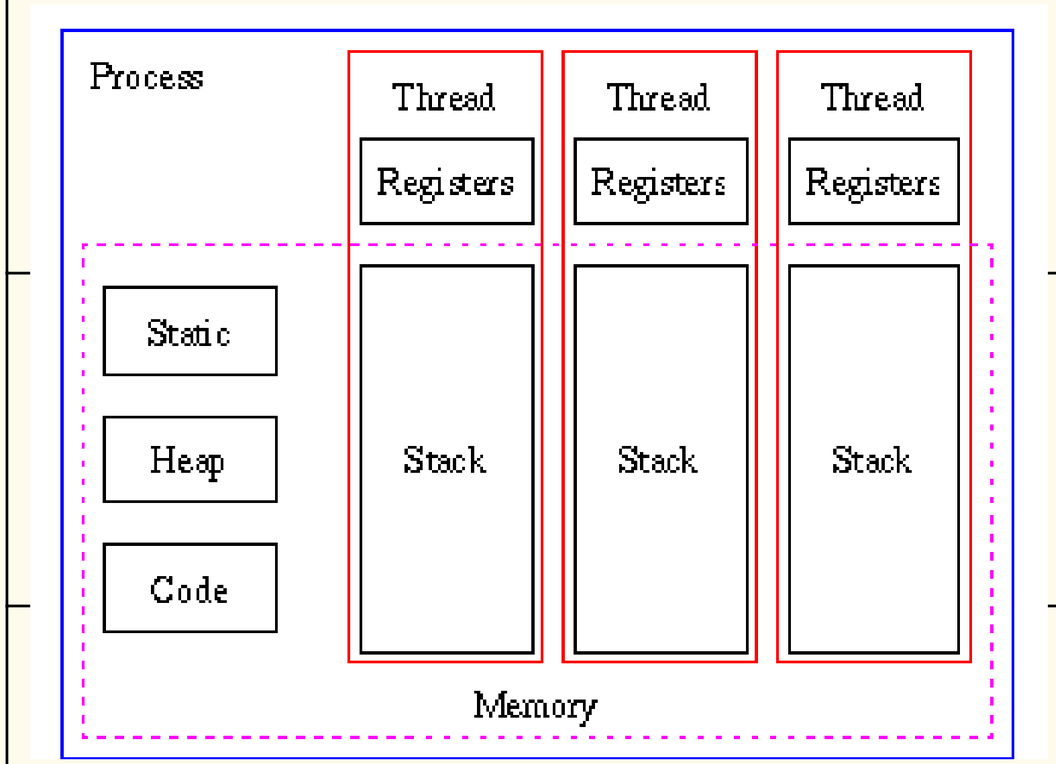
Thread

1	A part of execution within a process. A process contain one to many threads.
2	All threads in a process shares memory and resources.
3	Execution in a thread: 

Single-threaded VS Multithreaded

Execution of instructions in a single sequence.		Execution of multiple parts of a program at one time.
One thread in one process means one command can be processed at a time.		Many command process at the same time.

Multithreaded

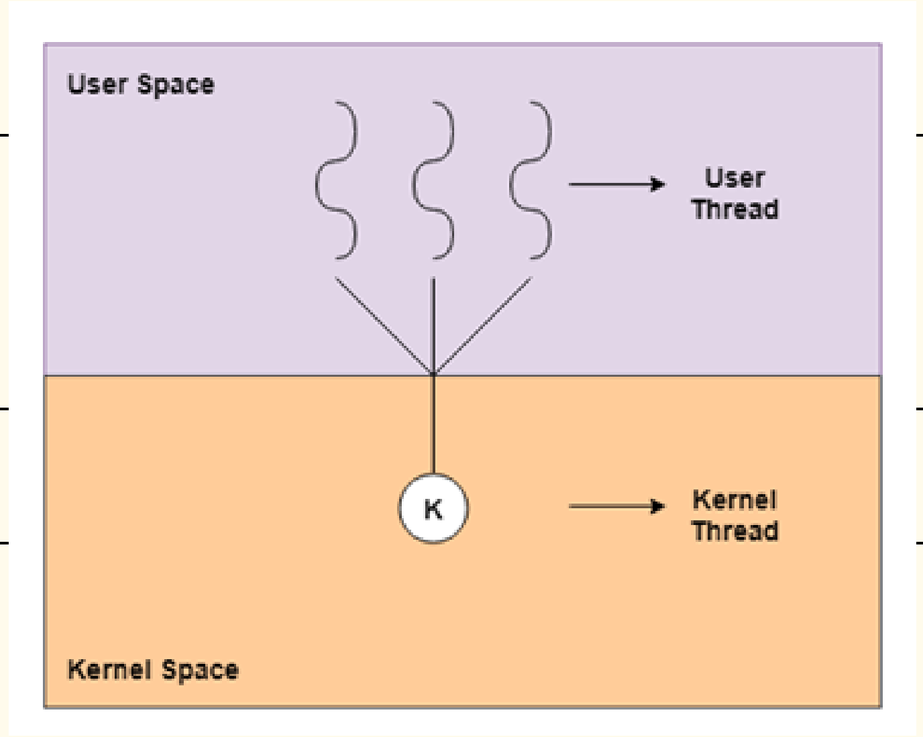


Two type of memory available: Stack and heap

Each thread has it own stack, but share the same heap.

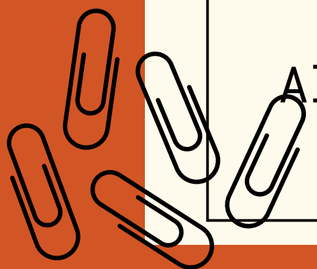
Weakness: If one thread have a problem, the others will effected.

Multithreaded Process Implementation: User-level Threads VS Kernel-level Threads

Implement by users only, not OS		Handle by OS directly by kernel
Implement as single-threaded process.		All threads manage by kernel
Small and faster		Slower
Performing blocking operation will block the entire process		Performing blocking operation not effect other thread.

Process VS Thread

An execution of program	Segment of process
Heavy-weight	Light-weight
More time for context switching	Less time for context switching
Not share memory with other process	Share memory with other threads in a process.
One process blocked, other can still execute.	User level thread blocked, all other threads effected.
All process run separately by OS.	All user level threads of same process run as a single task by OS.



DONE! THANK YOU :)

Group 1 members:
Madina Suraya, Nayli Nabihah, Irdina Aliah, Ameerul Habib