# OS Programming

*Programs and Processes*

Some of the content from today's slides comes from
https://www.geeksforgeeks.org/processes-in-linuxunix/
https://www.geeksforgeeks.org/wait-system-call-c/
https://www.geeksforgeeks.org/operating-system-process-table-process-control-block-pcb/

# Processes

Programs vs Processes

- A program is passive
  - A sequence of commands waiting to be run
- A process is active
  - An instance of program being executed
  - Can have multiple instance
  - There may be many processes running the same program
  - Also called job or task

# Processes

A process consists of all the services/resources that may be utilized during execution.

- Whenever a command is issued in unix/linux, it creates/starts a new process.
  - For example, when you run **grep**, a process starts.
- Unix/Linux keeps track of processes using a number called the process id or **pid**.
  - Each process in the system has a unique pid.
  - All pid's are unique.

# Process Management

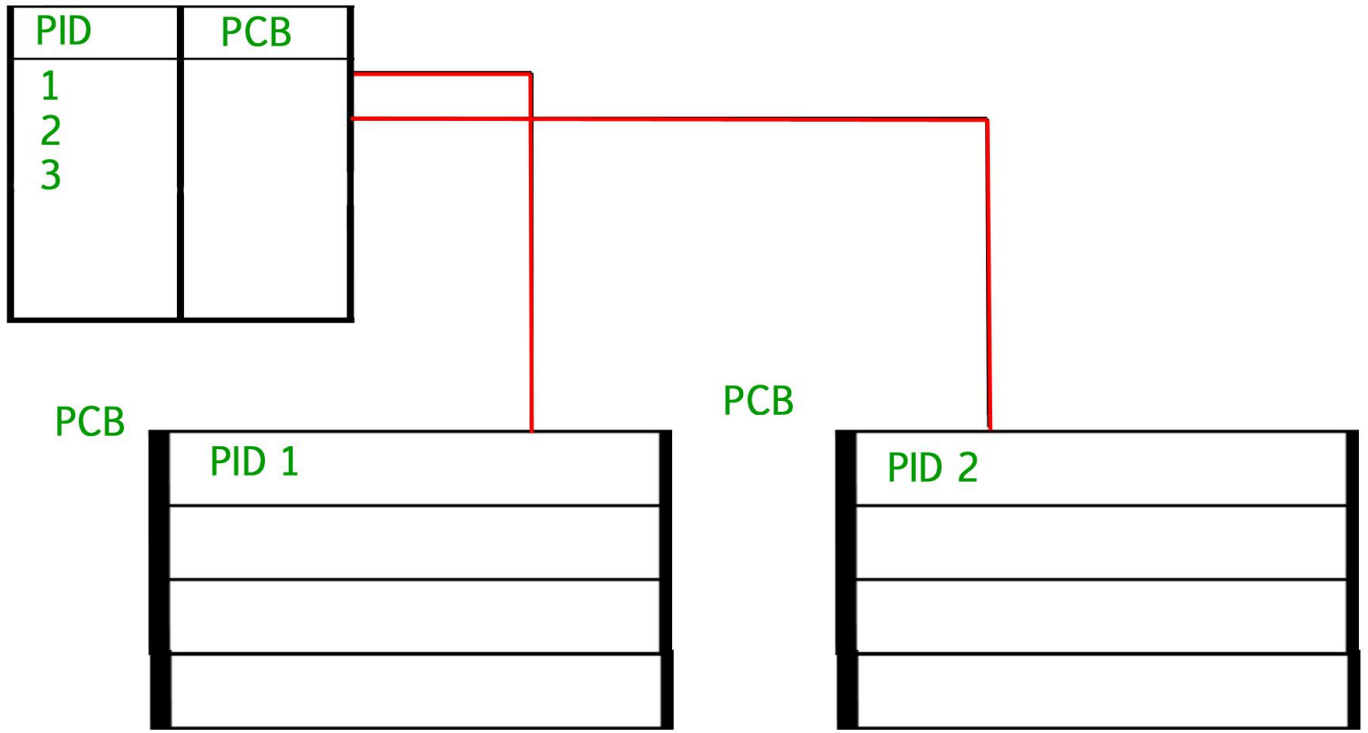
- Many processes run on the same machine at the same time, even on machines with a single CPU.
- An operating system manages processes and gives the illusion of concurrency.

Two main jobs of the OS in managing processes are:

- **Context Switching**, which consists of stopping one process and starting a new one.
- **Scheduling**, which consists of choosing a new process among the processes that are eligible for execution.

# Process Table

- The **process table** is a data structure maintained by the operating system to facilitate context switching and scheduling




Process table and process control block

# Process Management

- The **process table** is an array of **process control blocks**.
- A **process control block** (PCB) contains information about the process, i.e. registers, quantum, priority, etc. The process table is an array of PCB's, that means logically contains a PCB for all of the current processes in the system.

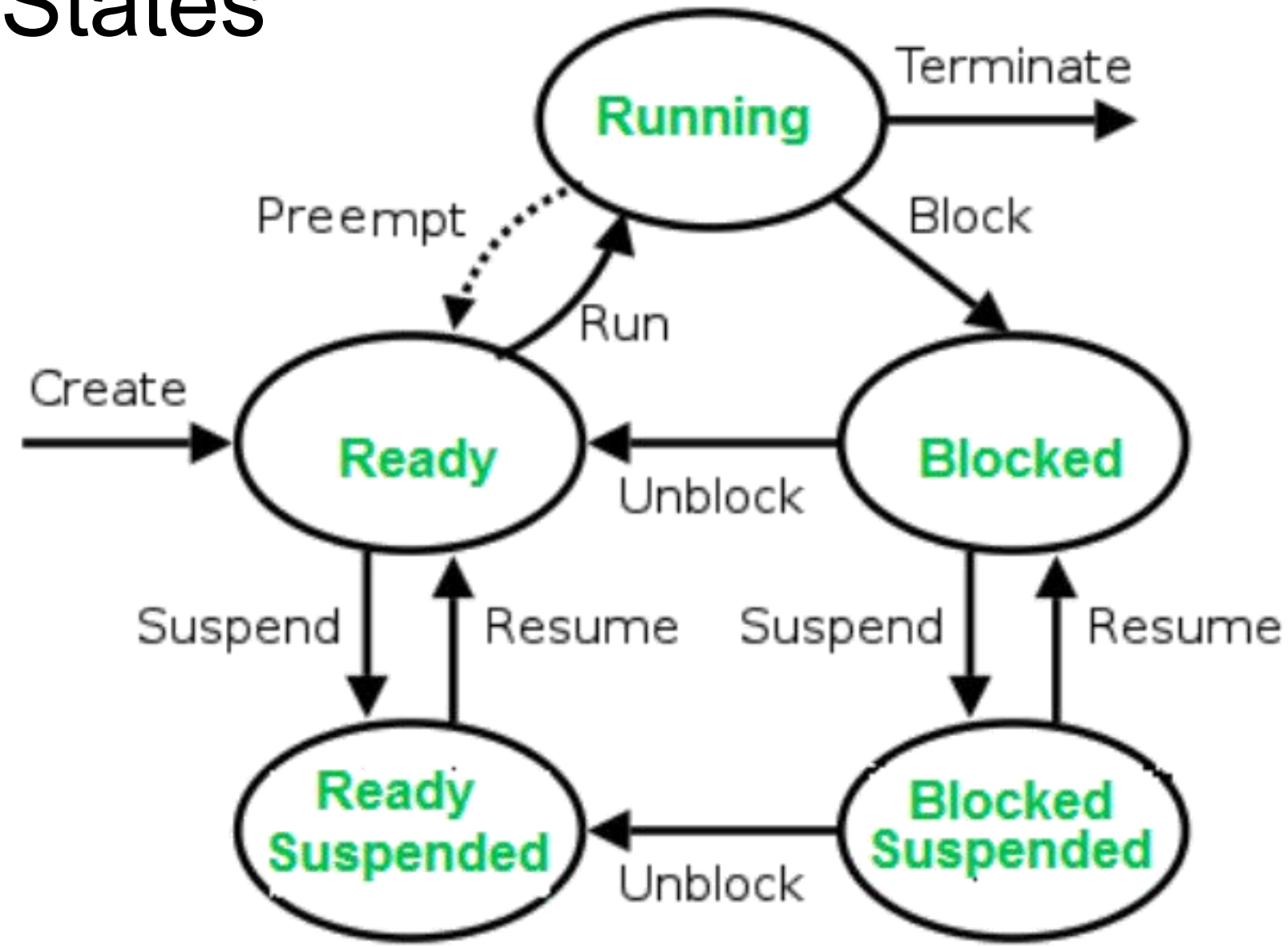

Process Control Block

# Process Control Block

- **Pointer** – It is a stack pointer which is required to be saved when the process is switched from one state to another to retain the current position of the process.
- **Process state** – It stores the respective state of the process.
- **Process number** – Every process is assigned with a unique id known as process ID or PID which stores the process identifier.
- **Program counter** – It stores the counter which contains the address of the next instruction that is to be executed for the process.
- **Register** – These are the CPU registers which includes: accumulator, base, registers and general purpose registers.
- **Memory limits** – This field contains the information about memory management system used by operating system. This may include the page tables, segment tables etc.
- **Open files list** – This information includes the list of files opened for a process.
- **Miscellaneous accounting and status data** – This field includes information about the amount of CPU used, time constraints, jobs or process number, etc.

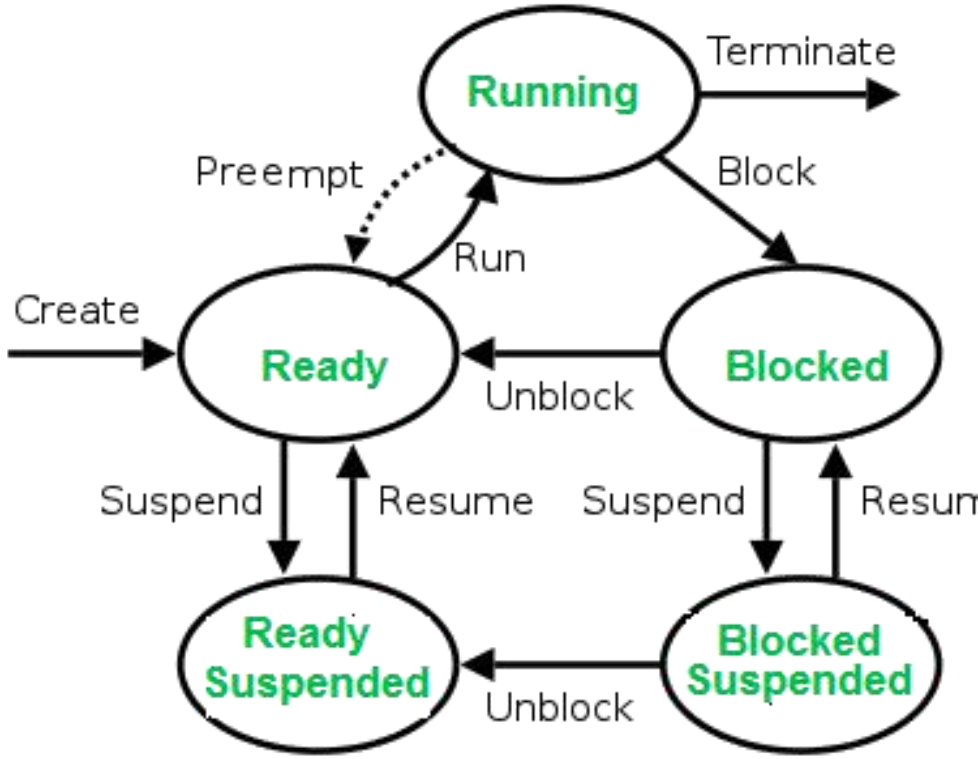

Process Control Block

# Process States



# Process States

- **New (Create)** – In this step, process is about to be created but not yet created, it is the program which is present in secondary memory (think hard drive) that will be read by OS to create the process.
- **Ready** – After creation of a process, the process enters the ready state i.e. the process is loaded into the main memory (think RAM). The process here is ready to run and is waiting to get CPU time for its execution.
- **Run** – The process is chosen by OS for execution and the instructions within the process are executed.
- **Blocked or wait** – Whenever the process requests access to I/O (e.g., needs an input from user) it enters the blocked or wait state. The process continues to wait in the main memory and does not require CPU. Once the I/O operation is completed the process goes to ready state.
- **Terminated or completed** – Process is killed as well as PCB is deleted.



# Process States

- **What happens when you run out of RAM?**
- **Suspend ready** – Processes that were initially in ready state but were swapped out of main memory and placed onto external storage by scheduler are said to be in suspend ready state. The process will transition back to ready state whenever the process is again brought onto the main memory.
- **Suspend blocked** – Similar to suspend ready but uses the process which was performing I/O operation and lack of main memory caused them to move to secondary memory. When work is finished it may go to suspend ready.