Process

Process –
A process is a program in execution. A process is the unit or work in a system.

- Process is an active entity while program is a passive entity.
- Current activity of process represented by the values of the program counter and other registers.

Process contains –
Text section - contains program code.
Data section – contains global variables.
Heap – memory that is dynamically allocated during process run time.
Stack – contains temporary data (such as local variable, return addresses, function parameters).

Process states –
Process state defines the current activity of the process. As a process runs, it goes through different states.

Each process may be in one of the following states-

- *New* - process is being created.
- *Running* – Instructions are being executed. Only one process can be running on processor at any instant.
- *Waiting* - process is waiting for its turn or waiting for some event to occur. Many processes may be in waiting state.
- *Ready* – process is ready for execution and waiting to be assigned to a processor. Many processes may be in ready state.
- *Terminated*- process has finished execution.

To know about structure of process in memory and process state diagram-

1. https://www.geeksforgeeks.org/gate-notes-operating-system-process-management-introduction/
2. https://www.geeksforgeeks.org/operating-systems-states-process/

Videos:

1. https://www.youtube.com/watch?v=GUt6o35euB0&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=3
2. https://www.youtube.com/watch?v=e8YxIdwodB8&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=4

Questions on Process State:

1. https://www.youtube.com/watch?v=3F0ir3tadLo&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=5

PCB (process control block) –
Each process is represented in system by a PCB which is also known as TCB (task control block). It contains piece of information associated with process including –

1. Process state: state may be new, running, waiting, exit and so on.
2. Program counter: this shows the address of next instruction to be executed for the process.
3. CPU registers.
4. Memory management information: it may include information like memory limits etc.
5. Accounting information: process number, time limits etc.
6. I/O status information: it includes the list of I/O devices allocated to process, list of open files, etc.

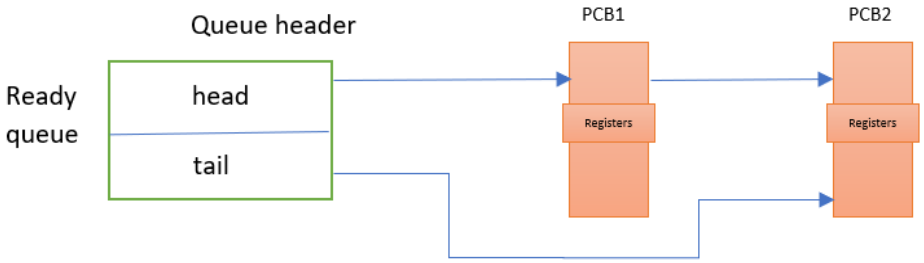
Reference - https://www.geeksforgeeks.org/operating-system-process-table-process-control-block-pcb/

Process scheduling-
The goal of multiprogramming is to maximize the CPU utilization and the goal of time sharing is to switch the CPU among processes so frequently so that user can interact with each program while it is running. So, to meet these objective process schedular come into picture. As process schedular selects a process from queue in some fashion and assigns it to CPU to execute. If a system has only one processor then there will never be more than one running process and other process will have to wait until CPU is free.

Scheduling queues:

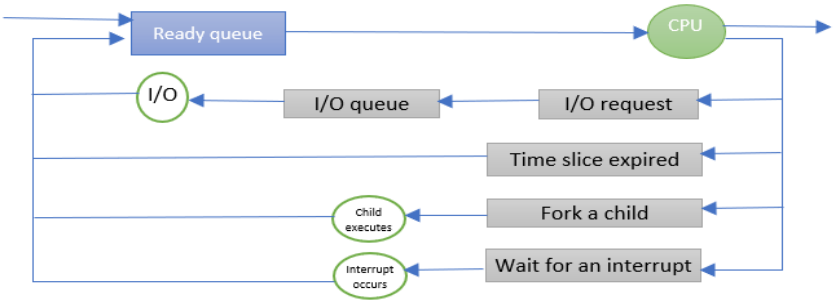
Job queue: It consists of all the processes in the system.

Ready queue: The processes that are residing in main memory and are ready but waiting to execute are kept in ready queue. This is stored as linked list. It contains queue header and queue tails.
Queue header contains pointers to first and last PCBs (process control block) in the list and each PCB points to next PCB in the ready queue.



Device queue: If the process makes an I/O request to a shared device and the device may be busy with the I/O request of some other process. So, the process may have to wait for the shared device. Therefore, device queue is the list of process waiting for a I/O device.

Queue Diagram: A common representation of process scheduling. It shows how the processes migrate among the various scheduling queues.



Here, two types of queues are present: ready queue and set of device queue.
Rectangle represents the queue.
Circles represent the resources that serve the queues.
Arrows represent the flow of processes in the system.

Process Schedulers:

The selection of process is carried out by the appropriate scheduler.

There are mainly three types of process scheduler:

1. **Long Term Scheduler**: It is also known as Job scheduler. It loads new processes into memory for execution. It controls the degree of multiprogramming i.e. the number of processes in memory. If the degree of multiprogramming is stable, then the average creation rate of processes is equal to average departure rate of processes. It is important that the long term schedular make a careful selection as most processes can be described as either CPU bound, or I/O bound.
   CPU bound process - spends more time in computations.
   I/O bound process - spends more time in doing I/O.

2. **Medium Term scheduler**: It helps in suspending and resuming the process as it mainly does swapping. Swapping helps in removing process from memory thus reducing degree of multiprogramming and later, the process can be reintroduced into memory and continue its execution from where it left off.

3. **Short Term Scheduler**: It is also known as CPU scheduler. It selects a process from ready queue and allocate the CPU to selected process.

Process scheduler in detail: https://www.geeksforgeeks.org/gate-notes-operating-system-scheduler/

Operations on Processes:

1. **Process Creation**:
   A process may create new process using system calls. The creating process is called a parent process and the new process is called the children of that process. New processes (children) may in turn also create other processes forms a tree of processes.
   Each process has unique id i.e. process id (pid) and each process is identified by its process id which is a unique integer. The value of pid for child process is 0 and for parent is integer value greater than 0.

   Fork() : It is a system call which is passed with no parameter and it is used to create a new process.
   Wait() : using this system call , the parent process wait for the child to complete and when the child process completes then parent process resumes from the call to wait(). Since parent process has nothing to do while child is running so parent process raises wait () system call to move itself from ready queue until the termination of child.

   When process creates new process then different possibilities exists –
   a. In terms of execution –
      - The parent process continues to execute with its children.
      - The parent process waits until its children have terminated.

   b. In terms of address space –
      - The child process may be duplicate of parent process.
      - The child process may have a new program loaded into it.

2. **Process termination**:
   A process terminates when it finishes its execution and ask the operating system to delete it (deletion of PCB) by using the exit() system call.

   - **Orphan process**: The operating system doesn't allow child process to continue if its parent terminates. So, this type of child process is known as orphan process.

   If a parent process terminates either normally or abnormally then all its children must also be terminated so this type of phenomenon is known as cascading termination.

   ***Note** - when parent process wait for the child process termination using wait() system call then this wait() system call returns the process identifier (pid) of a terminated child process so that parent process can determine which of its child has terminated.

   - **Zombie process**: If a process is terminated but its PCB still exists as its parent process has not yet accepted its return value.
     Learn more about zombie process and prevention - https://www.geeksforgeeks.org/zombie-processes-prevention/

Interprocess Communication:

Types of process: Independent process and cooperating process.

1. Independent process: It cannot affect or be affected by other processes executing in system.
2. Cooperating process: it can affect or be affected by other processes executing in the system. It shares data with other processes.

Interprocess communication (IPC): It provide a mechanism which allow processes to exchange data or information.

Models of IPC:

a. Shared Memory.
b. Message passing.

a. Shared Memory:

In this region of memory is established and then processes can exchange information through this shared region. Process can read or write data to the shared region.
E.g. Producer consumer problem – in this producer process produces the data that is consumed by consumer process or we can say that, Producer process filled the shared memory that is emptied by consumer process.

b. Message passing:

Communication is done by means of messages. In this, messages are exchanged between the cooperating processes. It uses two operations: send(message) and receive(message).

Message communication can use either direct communication or indirect communication

1. **Direct communication**: explicitly name the sender and receiver of the communication.
   - send (A, message): send message to process A.
   - receive (B, message): receive message from process B.
2. **Indirect communication**: In this, messages are sent to the object (mailbox or ports) and process receive messages from that object.
   - send (A, message): send message to object A.
   - receive (A, message): receive message from object A.

Comparison between Shared memory and Message passing:

Shared Memory	Message Passing
It is helpful when process wants to share large amount of data.	It is helpful when process wants to share small amount of data.
In this system calls are required only to establish shared region.	Message passing system typically implemented using system calls.
It allows maximum speed for communication.	It is not fast as compare to Shared memory because it requires more time-consuming task of kernel.

Please read these links –

1. https://www.geeksforgeeks.org/inter-process-communication/
2. https://www.geeksforgeeks.org/interprocess-communication-methods/

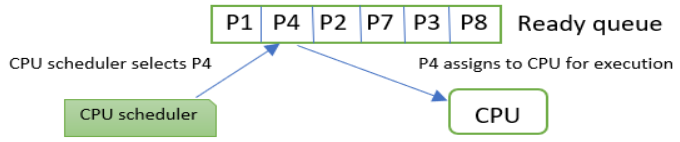
Process Scheduling:

Before start with process scheduling, first explore these links in the given sequence:

1. Introduction: https://www.geeksforgeeks.org/gate-notes-operating-system-process-management-introduction/
2. Process stated: https://www.geeksforgeeks.org/operating-systems-states-process/
3. Process schedulers: https://www.geeksforgeeks.org/gate-notes-operating-system-scheduler/

CPU scheduler:

CPU scheduler can select any process in some fashion but not necessarily in FIFO.
Ready queue can be implemented as priority queue, unordered list, FIFO queue etc.
All the processes that are in ready queue and are ready to execute but waiting for a chance to run on the CPU.



Dispatcher: It is a module that give control of the CPU to process selected by CPU scheduler (short term scheduler).
Difference between dispatcher and scheduler : https://www.geeksforgeeks.org/operating-system-difference-dispatcher-scheduler/

Preemptive scheduling: Running process is interrupted for some amount of time and resumed later when the priority task has finished its execution**.** There is an overhead of context switching. It is done under two circumstances –

1. When a process switches from the waiting state to the ready state (completion of I/O).
2. When a process switches from the running state to ready state (when interrupt occurs).

Non-Preemptive scheduling: Running process is executed till its completion. It is done under two circumstances -

1. When a process switches from running state to the waiting state.
2. When process terminates.

Difference between preemptive and non-preemptive scheduling : https://www.geeksforgeeks.org/preemptive-and-non-preemptive-scheduling/

Important parameters in Process scheduling:

1. **CPU utilization:** When we want to keep CPU as busy as possible.
2. **Throughput:** Number of processes that are executed pe unit time.
3. **Arrival time (AT):** Time at which the process enters the ready state.
4. **Burst time (BT):** Total amount of time that process spends in running state.
5. **Completion time (CT)**: Time at which process get finished.
6. **Waiting time (WT):** total amount of time that process spends in waiting state
7. **Turnaround time (TAT):** The difference between the time of arrival and the time of completion.
   TAT = CT – AT OR TAT = BT + WT
8. **Response time:** the first time at which the process hits the CPU.

Scheduling Algorithms:

1. **Before starting this, first go through this link:** https://www.geeksforgeeks.org/gate-notes-operating-system-process-scheduling/

Now, you have some basic idea about scheduling algorithms. Let's learn them in detail.

A. **First Come First Serve (FCFS)** –
The process that request CPU first is allocated the CPU first. It works in FIFO manner. Its criteria based on Arrival Time and the mode is non preemptive (process holds the CPU till it finishes). When the CPU is free, it is allocated to the process that is selected from the head of ready queue.

Theory:

1. https://www.geeksforgeeks.org/program-fcfs-scheduling-set-1/
2. https://www.geeksforgeeks.org/program-fcfs-scheduling-set-2-processes-different-arrival-time/

Numerical on FCFS:
https://www.youtube.com/watch?v=HIB3hZ5fHw&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=8

Advantages of FCFS:

1. It is simple and doesn't include any complex logic.
2. Every process gets a chance to run.

Disadvantages of FCFS:
1. Throughput is not efficient.
2. Convoy effect.

Convoy effect: If a process having large burst time then all the processes which are after it are going to starve i.e. large burst time process starves the small burst time processes.
Convoy effect explanation with the help of diagram: https://www.geeksforgeeks.org/convoy-effect-operating-systems/

B. Shortest Job first scheduling (SJF):
When CPU is free then it is allocated to the process that has smallest burst time. Its criteria based on Burst Time. Whenever two processes have same burst time then FCFS scheduling is used i.e. the process with the least arrival time is chosen.

Theory:
1. In Non preemptive mode: https://www.geeksforgeeks.org/program-shortest-job-first-sjf-scheduling-set-1-non-preemptive/
2. In preemptive mode: https://www.geeksforgeeks.org/program-shortest-job-first-scheduling-set-2srtf-make-changesdoneplease-review/
3. SJF with predicted burst time: https://www.geeksforgeeks.org/operating-system-shortest-job-first-scheduling-predicted-burst-time/

Numerical: https://www.youtube.com/watch?v=xh8hRjtbu3k&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=12
Analysis of SJF:
https://www.youtube.com/watch?v=pknLjapNc2g&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=13

SJFS with predicted Burst time:
https://www.youtube.com/watch?v=8SLj1Z4ness&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=15

Advantages of SJF scheduling:
1. Shorted burst time processes are executed first followed by large burst time processes.
2. Throughput is increased.
3. Minimum average waiting time and turnaround time.
4. It is useful in performance measuring.

Disadvantages of SJF scheduling:
1. It is not practical to implement.
2. Starvation to larger burst time processes.

C. **Priority scheduling**:
Priority is associated with each process and the CPU is allocated to process having highest priority. In this, equal priority processes are scheduled in FCFS fashion. SJF is the special case of priority scheduling in which smaller the CPU burst time, higher the priority.

Priorities can be defined internally or externally. Priority scheduling can be preemptive or non-preemptive.
Preemptive priority scheduling: this algorithm preempts the CPU if the priority of newly arrived process is higher than the priority of running process.
Non-Preemptive priority scheduling: it simply put the newly arrived process at the head of the ready queue.

Theory:

1. Priority scheduling: https://www.geeksforgeeks.org/program-priority-scheduling-set-1/
2. Priority scheduling with different arrival time: https://www.geeksforgeeks.org/operating-system-priority-scheduling-different-arrival-time-set-2/
3. Introduction video: https://www.youtube.com/watch?v=Iw0XyasWxDk&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=24

Numerical:

1. Non Preemptive priority scheduling: https://www.youtube.com/watch?v=i4PQucowf1c&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=25
2. Preemptive priority scheduling: https://www.youtube.com/watch?v=hDn4hM148V8&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=26

Advantage:

1. Priority of process can be selected on various factors like user preference, memory requirement, time requirement.

Disadvantages:

1. Indefinite blocking or starvation may be possible.
2. It may leave some low priority processes waiting indefinitely.

Aging: It is the solution for indefinite blocking of low priority processes. It gradually increases the priority of process that wait for a long time.
Starvation and Aging : https://www.geeksforgeeks.org/starvation-aging-operating-systems/

D. Round Robin Scheduling (RR):

It uses the time quantum. It executes process for a time quantum. Every process gets fair chance after some amount of time i.e. quantum time or time slice. This time quantum is generally from 10 to 100 milliseconds. It is the combination of FCFS and preemption algorithm. It is designed specially for time sharing systems.

Theory:

1. https://www.geeksforgeeks.org/program-round-robin-scheduling-set-1/
2. https://www.geeksforgeeks.org/operating-system-selfish-round-robin-scheduling/
3. https://www.geeksforgeeks.org/round-robin-scheduling-with-different-arrival-times/

Numerical:

1. Example 1: https://www.youtube.com/watch?v=Sa600YsU16U&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=16
2. Example 2: https://www.youtube.com/watch?v=Q3Clg_aAQtw&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=17
3. Example 3: https://www.youtube.com/watch?v=v9PbedfdrVE&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=18
4. Example 4: https://www.youtube.com/watch?v=g2vqTYUimTo&list=PLGvfHSgImk4Y6htSDkXHZWTaC99rz_ApY&index=19

Advantages:

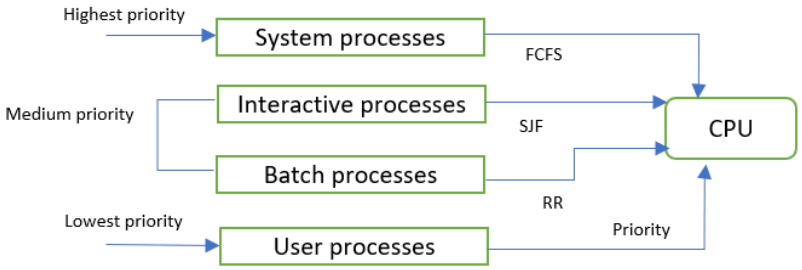
1. Every process gets a fair chance to run.
2. CPU is shared between all processes.

3. Starvation doesn't occur.

Disadvantages:

1. High priority task may not execute the full instruction in given time quantum.
2. Throughput mainly depends on the length of time quantum.
3. If the time quantum is too small than needed, CPU switching time from one process to other is increased that leads to decrease in CPU efficiency.

E. **Multilevel queue scheduling**:
It is very difficult to schedule all the processes using one queue as there are various types of processes in the system. So, for this we use multilevel queue. Multilevel queues divide the ready queue into several separate parts. Each queue has its scheduling algorithm. Order of execution starts from high priority queue, if high priority queue is emptied then medium priority queue gets a chance to executer and if medium priority queue emptied then lower priority queue get a chance to execute.



System processes: Processes which belongs to the OS.
Interactive process (foreground): process that interacts with the computer e.g. games etc.
Batch processes(background): multiple jobs are submitted to the computer and computer executes them one after other, without any user interactions.

Whenever system processes are there then no other process will be taken for scheduling as system process have highest priority as compare to others. In this, all the high priority processes are executed first, when high priority processes execute completely then low priority processes gets a chance to execute. So, it may lead to starvation to low priority processes.

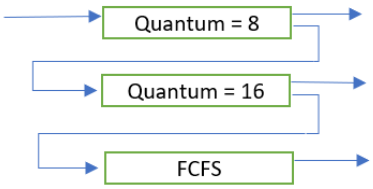
Working : https://www.geeksforgeeks.org/operating-system-multilevel-queue-scheduling/
Video: https://www.youtube.com/watch?v=1w9FybdNi_Y

Advantage: we can apply separate scheduling algorithm for different type of process.

Disadvantage:

1. Categorization of processes needs to be done before process enters the system.
2. Starvation to lower level queue.

F. **Multilevel feedback queue scheduling**:
Unlike multilevel queue scheduling where processes are permanently allocated to queue when they enter the system, multilevel feedback queue allows a process to move between queues. This movement give chance to lower priority processes. It uses the concept of Aging. Here, we have various levels of queue and after some time process moves from lower level priority queue to upper level and the process will go on so that at highest priority queue process gets CPU and executed. If process have too much CPU burst time, then it will move to low priority queue.



Multilevel feedback queue is defined by several parameters:

1. The number of queues.
2. The scheduling algorithm for each queue.
3. The method used to determine when to move a process to higher priority queue.
4. The method used to determine when to move a process to lower priority queue.

Theory: https://www.geeksforgeeks.org/multilevel-feedback-queue-scheduling/
Video: https://www.youtube.com/watch?v=1w9FybdNi_Y

Advantage:

1. Give preference to short jobs.
2. Categorize the process based on their need for the processor.
3. No process gets starve for a long time because it is going to move up.

G. Longest Job First (LJF)

In this, Process having longest Burst time gets scheduled first. It is non preemptive in nature but preemptive mode of longest job first is known as Longest Remaining Time First (LRTF).

Numerical:
https://www.youtube.com/watch?v=xsOjx2Ih1-s&list=PLEbnTDJUr_If_BnzJkkN_J0Tl3iXTL8vq&index=20

Disadvantage:

1. Small burst time process may starve for CPU.

H. Longest Remaining Time First (LRTF):

The preemptive mode of longest job first (LJF) is known as LRTF. In this, run the process till other process become longer.
Theory and numerical:

1. https://www.geeksforgeeks.org/cpu-scheduling-longest-remaining-time-first-lrtf-program/
2. https://www.geeksforgeeks.org/cpu-scheduling-longest-remaining-time-first-lrtf-algorithm/

videos:

1. LJF: https://www.youtube.com/watch?v=xsOjx2Ih1-s&list=PLEbnTDJUr_If_BnzJkkN_J0Tl3iXTL8vq&index=20
2. LRTF: https://www.youtube.com/watch?v=QoDe5B2x8bo&list=PLEbnTDJUr_If_BnzJkkN_J0Tl3iXTL8vq&index=21
3. Gate question on LRTF: https://www.youtube.com/watch?v=5DlzEI5vfdA&list=PLEbnTDJUr_If_BnzJkkN_J0Tl3iXTL8vq&index=22

I. Highest Response Ratio Next (HRRN):

Its Criteria is based on the Response ratio and the mode is non preemptive. It not only favors the shorter burst time processes but also limits the waiting time of longer burst time processes.

Response Ratio = (W + S) / S
Where, W = waiting time of process.
S = Burst time or service time for a process.

Theory: https://www.geeksforgeeks.org/operating-system-highest-response-ratio-next-hrrn-scheduling/

Video: https://www.youtube.com/watch?v=ABKa32ZS0Os&list=PLEbnTDJUr_IE_BnzJkkN_J0TI3iXTL8vq&index=23

J. **Lottery Process scheduling**:
This scheduling is different from all other scheduling. It is the probabilistic scheduling algorithm for processes. In this, processes are scheduled in random manner as each process is assigned with some number of lottery tickets and the scheduler draws a random ticket to select the next process. This scheduling solved the problem of starvation as each process have lottery ticket which shows each process has non-zero probability of being selected at each scheduling operation.

Lottery process in detail: https://www.geeksforgeeks.org/operating-system-lottery-scheduling/

K. Multiprocessor scheduling:

1. https://www.geeksforgeeks.org/operating-system-multiple-processor-scheduling/