OPERATING SYSTEMS

#5

After A.S.Tanenbaum, Modern Operating Systems, 3rd edition

Uses content with permission from Assoc. Prof. Florin Fortis, PhD

PROCESS SCHEDULING

General information

PROCESS SCHEDULING GENERAL INFORMATION

Cooperating processes

- A process is independent if it cannot affect or be affected by other process (in execution).
- A process that is not independent is a cooperating process.
- Usually, cooperating processes share several resources (like PIPE files, memory areas and others) with other processes.
- Cooperation is needed in order to:
  - Share information.
  - Obtain a better computing speed.
  - Offer modularity, eventually by splitting elementary tasks between different processes.
  - Offer convenience, in order to solve several tasks simultaneously.

PROCESS SCHEDULING GENERAL INFORMATION

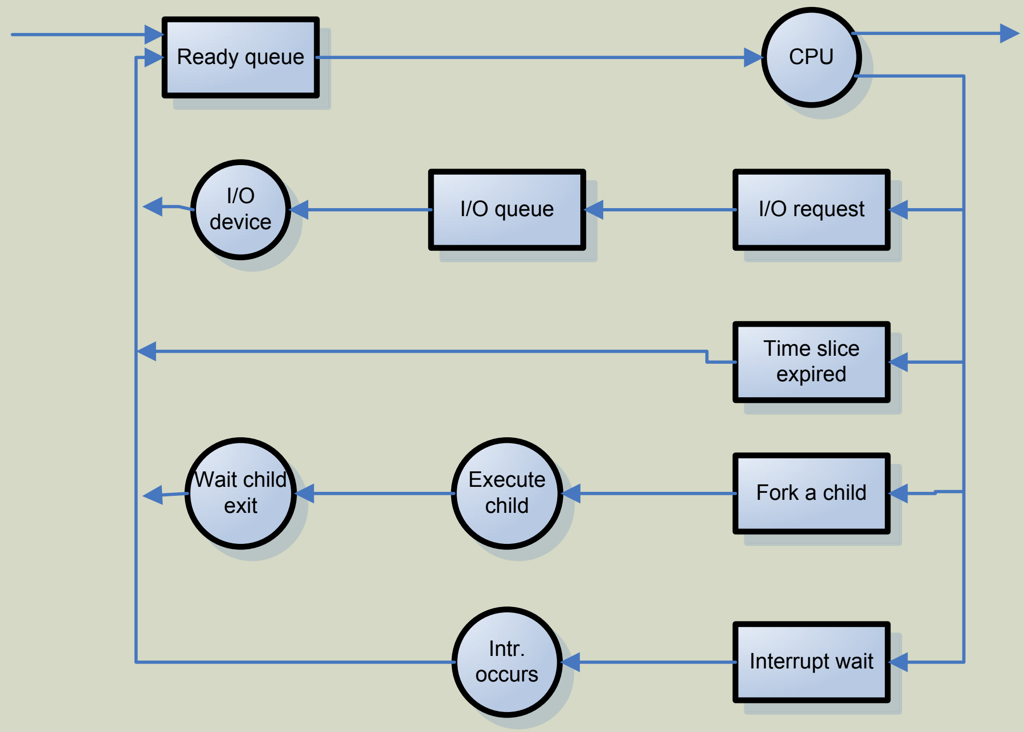
- Multiprogramming’s objective: Maximizing processor usage
  - Time-shared systems are using a frequent switch of the computing resources between the different processes;
  - User interaction is based on this mechanism.
- For simple, mono-processor systems, multiprogramming and time-sharing are similar
  - the processor is offered alternatively to processes by a frequent switch between them.

PROCESS SCHEDULING GENERAL INFORMATION

Scheduling queues

- Processes are always in one of system's queue:
  - JOB QUEUE: Processes accepted in the system;
    - This queue offer the entire pool of existent jobs.
  - READY FOR EXECUTION QUEUE: processes that are ready for execution;
    - These processes are stored in the memory (if possible, all ready processes are stored in this queue).
    - The BCP must contain a specialized field for ready-queue support.
    - Processor scheduling is highly based on this particular queue.
  - I/O QUEUE & INTERRUPT QUEUE: blocked processes
    - Also blocked processes could be in a queue for additional execution time.
- Processes migrate between the different queues according to the underlying process model.

PROCESS SCHEDULING
SCHEDULING QUEUES



- Process migration between the different queues.
- Not specified: queues related with mid-term scheduling (swapping/ virtual memory)

PROCESS SCHEDULING GENERAL INFORMATION

Schedulers. Levels of scheduling

- Processes stored in different queues are competing for accessing different resources.
  - The entire existence of a process is based on a permanent migration between these queues.
- The OS, by its specialized components (schedulers), is responsible with choosing the processes that are going to access these resources.
- This choice is based on a scheduling algorithm, and must assure a good balance between the different existent processes:
  - The multiprogramming level of the system must be maintained at the same level, if possible.
    - the number of processes simultaneously stored in memory at a time
- The system should offer two major categories of processes
  - Limited to I/O operations
    - Spend most of the time to solve input/output operations instead of computation.
  - Limited to processor
    - Spend most of the time in order to do some computations, instead of I/O operations.

PROCESS SCHEDULING
GENERAL INFORMATION

- *Scheduling (planning) decisions* are helping the operating system
  - OS is helped for taking important decisions, related with resources usage.
  - These decisions are based on *scheduling algorithms*, affecting the activity of the entire system.
- Categories of scheduling algorithms
  - Batch – quite simple, but effective
  - Interactive
  - Real-time – quite complex.

PROCESS SCHEDULING GENERAL INFORMATION

- For different kinds of OS, there are diverse needs, arising from the special existing requirements.
  - For an operating system for micro-computers the OS doesn't care which of the processes is to be scheduled next while there is a reasonable response for its application.
- There are different needs for systems with a high load, like server OS.
- The activity of a scheduler is highly influenced by another important activity in the OS: *the context (or process) switch*.
  - Being a costly activity, it must be carried out in an efficient manner.

PROCESS SCHEDULING GENERAL INFORMATION

Categories of processes.

- Two major categories of processes are to be considered during the planning activities:
  - I/O bounded (limited) processes;
  - Processor bounded (limited) processes.

Scheduling moments.

- Scheduler decisions should be made, from different reasons, at different moments in the lifecycle of a process.
- These moments include:
  - Creation of a new process;
  - Termination of a process;
  - An interrupt for some I/O request;
  - Blocking/suspending of processes;
  - Occurrence of a trap (e.g. timer).

PROCESS SCHEDULING
GENERAL INFORMATION

Based on scheduling time, there are two important categories of algorithms:

- **Non-preemptive algorithms.**
  - Algorithms used when process activity cannot be interrupted.
    - Batch-processing systems are preferring these algorithms.
    - Eventually, they are using preemptive algorithms with very large limits of time.
- **Preemptive algorithms.**
  - Processes are chosen for execution for a limited period of time.
    - Interactive systems are preferring these algorithms.
    - Observe that a non-preemptive algorithm is not suitable for an interactive behavior.

PROCESS SCHEDULING GENERAL INFORMATION

Requirements

- **Fairness**: Comparable processes should be viewed in the same way.
  - They should be given the same chance to reach resources.
- **Equilibrium**: scheduling activities should offer a high level of usage for most of the components of a computing system.
  - This should offer a higher global productivity.
  - For example, one should avoid to prefer processor bounded processes instead of I/O bounded processes.
- **Obey policies**: the scheduling algorithm should adapt itself to the local policy of the system, imposed by the specific requirements of the computing system.

PROCESS SCHEDULING GENERAL INFORMATION

All systems

Fairness - giving each process a fair share of the CPU
Policy enforcement - seeing that stated policy is carried out
Balance - keeping all parts of the system busy

Batch systems

Throughput - maximize jobs per hour
Turnaround time - minimize time between submission and termination
CPU utilization - keep the CPU busy all the time

Interactive systems

Response time - respond to requests quickly
Proportionality - meet users' expectations

Real-time systems

Meeting deadlines - avoid losing data
Predictability - avoid quality degradation in multimedia systems

PROCESS SCHEDULING
GENERAL INFORMATION. BATCH

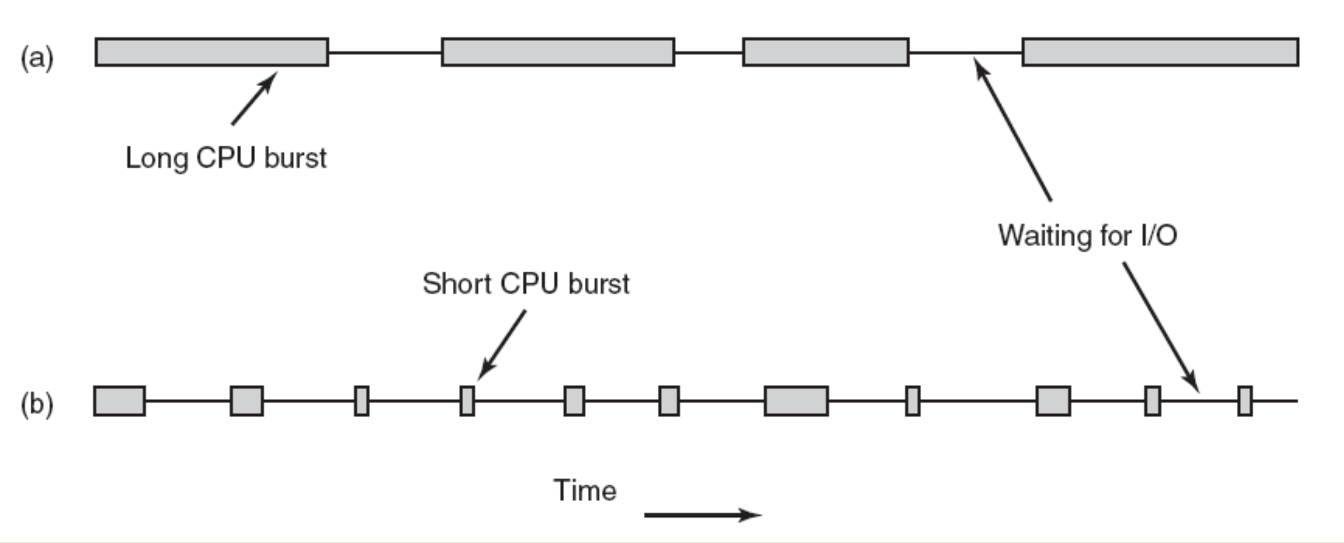
- For batch scheduling algorithms, one will follow the evolution of several parameters:
  - **Throughput**: the number of jobs processed in the unit of time (for example, one hour).
  - **Return time**: for a process, this is the difference between the moment of job termination and the moment of job transmission.
    - For the algorithm, the mean value of return times.
  - **Processor usage**: requirement imposed by the high costs of computing systems of this type.

PROCESS SCHEDULING
GENERAL INFORMATION. INTERACTIVE

- For interactive scheduling algorithms, one will follow the evolution of several parameters:
  - **Response time**: the difference between the moment of transmission and the moment of obtaining the results.
  - **Proportionality**: waiting time should be kept in reasonable limits, acceptable by the end user.
  - **Waiting time**: this is the sum of all periods of inactivity for a process.
    - One can compute the medium waiting time, starting from these values.

PROCESS SCHEDULING GENERAL INFORMATION.

Bursts of CPU usage alternate with periods of waiting for I/O.



(a) A CPU-bound process.
(b) An I/O-bound process.

PROCESS SCHEDULING
GENERAL INFORMATION. ALGORITHM EVALUATION

- While evaluating different scheduling algorithms, one will follow several parameters (even if some of them are not relevant), including:
  1. The average ***response*** (or ***return***) ***time***;
  2. The ***average waiting time***;
  3. ***Arrival time***; the moment a process is accepted in the system.
  4. "***Burst***" ***time***; the duration of a typical execution cycle (see previous slide).
  5. ***Context switches***; total number of context (process) switch operations needed.

PROCESS SCHEDULING

Algorithms. Batch

PROCESS SCHEDULING BATCH. ALGORITHMS (1)

- FCFS (First Come First Served), also FIFO
- The simplest scheduling algorithm.
  - Processes are allocated to the processor in their arrival order.
  - All processes are in the same waiting queue.
  - Every process that is interrupting (temporarily) its activity is accepted again at the end of the waiting queue.
  - It is a **non-preemptive algorithm**.

PROCESS SCHEDULING BATCH. ALGORITHMS (2)

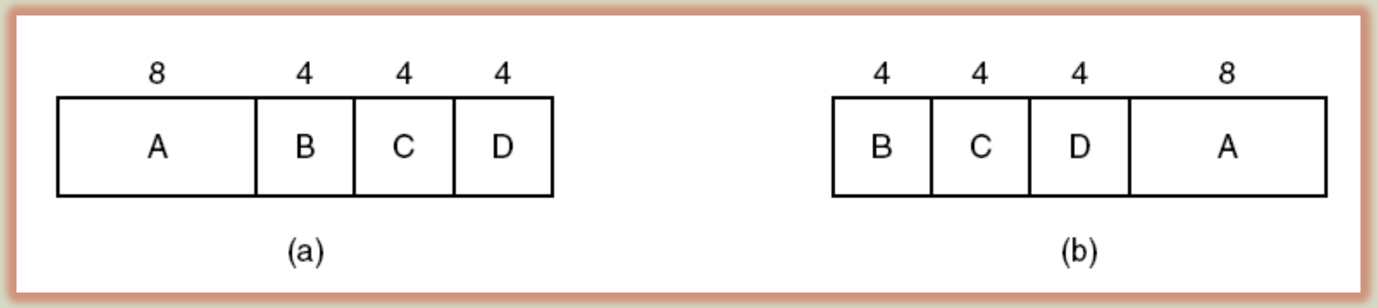
- Shortest Job First (SJF)
  - Also a **non-preemptive algorithm**, based on the (estimated) duration of a process.
  - First, the process with the shortest estimated execution time is chosen.
  - When several processes have the same estimated execution time, FCFS (FIFO) algorithm is used.

PROCESS SCHEDULING BATCH. ALGORITHMS (3)

- **Shortest Job First (SJF)**
  - This algorithm is optimal for the average waiting time. Unfortunately, it is quite difficult to obtain a good estimation for the execution time.
  - The SJF algorithm can be used rather for long term scheduling.
  - Usually, estimated execution time is obtained starting from the idea that it is similar with previous durations.

PROCESS SCHEDULING BATCH. ALGORITHMS (4)

Shortest Job First



(a) Running four jobs in the original order.
(b) Running them in shortest job first order.

PROCESS SCHEDULING BATCH. ALGORITHMS (5)

- Shortest Return Time Next
  - This algorithm is similar with SJF, but this time we are using the estimated time needed for process termination.
  - A scheduler decision should be made when a new process is started or when a process is terminating its activity.
  - This is a preemptive algorithm (usually, it is named the preemptive SJF algorithm). Also known as “shortest remaining time first” algorithm.

PROCESS SCHEDULING

Algorithms.
Interactive

PROCESS SCHEDULING INTERACTIVE. ALGORITHMS (1)

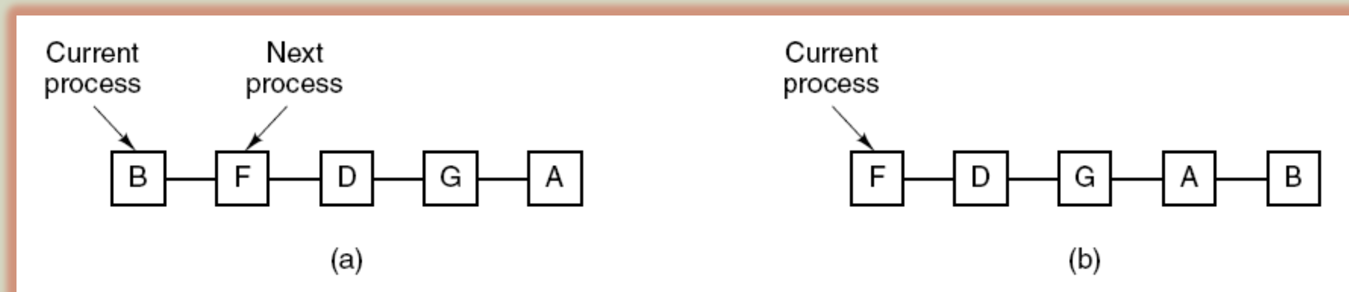
- Round Robin (RR)
  - This is one of the oldest scheduling algorithms used for interactive systems (and still in use).
  - For every process there is a quantum of time, the maximum time slice that a process is allowed to use for an execution cycle.
  - A process with an expired time slice is preempted and put at the end of ready-processes list.
  - The chosen quantum should be as little as possible, such that the time “lost” by the processor with context switch operations to be as little as possible.

PROCESS SCHEDULING INTERACTIVE. ALGORITHMS (2)

- Round Robin (RR)
  - The performance of the entire algorithm highly depends on the value of the quantum. For a very high value, the algorithm is similar with the FCFS algorithm.
  - If the value is too small, the RR approach is called “processor sharing”.
  - The value of the quantum should cover, as much as possible, 80% of the estimated execution times. In this way, at most 20% of the processes are going to request context switches.
  - The response time is influenced by this choice: one will obtain a better value when most of the processes need at most the time of a quantum for their execution.

PROCESS SCHEDULING INTERACTIVE. ALGORITHMS (3)

Round Robin Scheduling



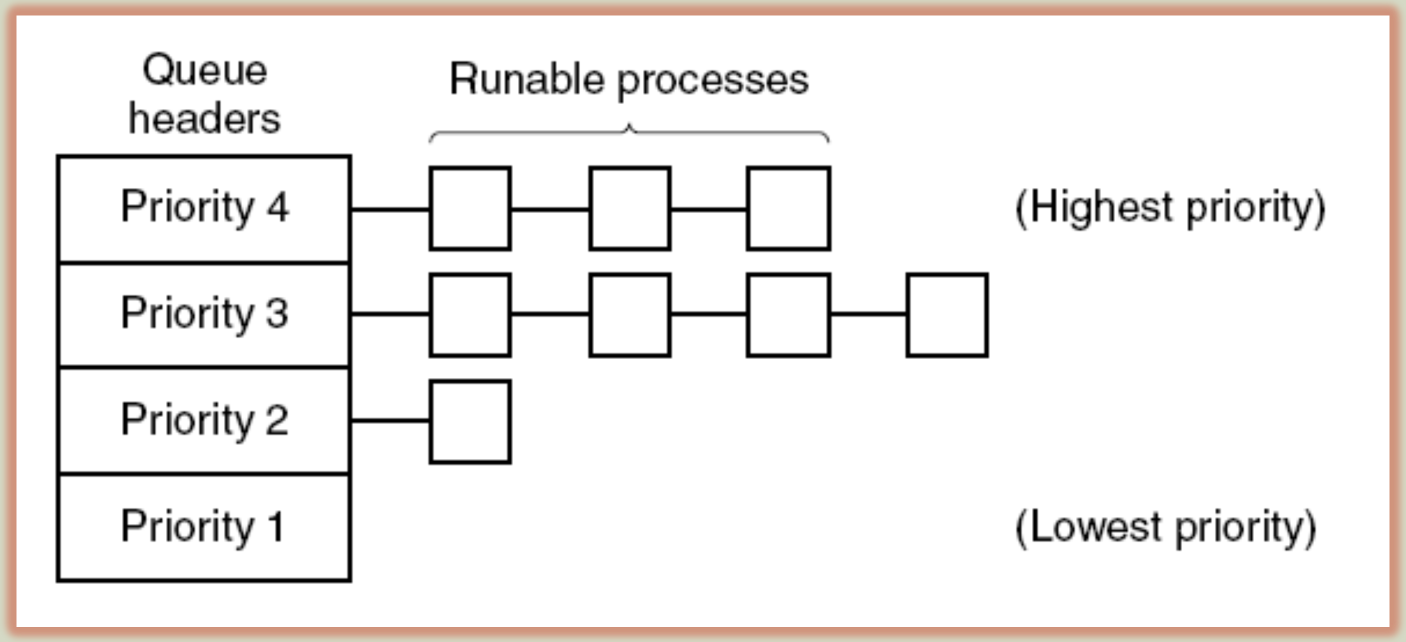
(a) The list of runnable processes.
(b) The list of runnable processes after B uses up its quantum.

PROCESS SCHEDULING INTERACTIVE. ALGORITHMS (4)

- Priority Scheduling
- Now processes are no longer with the same importance.
  - This time, every process is characterized by a value (its priority) that is used in order to determine the next process chosen for execution.
- Processes with higher priorities are always preferred for scheduling.
  - In order to avoid a monopoly from these processes, the scheduler could periodically modify the priorities of old processes or of the processes that are waiting for a long period of time.
- The priorities scheme is highly influenced by the policy of the system.
- Several schemas can be used in order to dynamically determine the priorities, in order to have the same chances of execution.
- An acceptable version:
  - scheduling based on priority classes. In this case the entire class of processes is chosen. Internal scheduling can be made by using another algorithm, like the RR algorithm.

PROCESS SCHEDULING INTERACTIVE. ALGORITHMS (5)

Priority Scheduling



PROCESS SCHEDULING INTERACTIVE. ALGORITHMS (6)

- Shortest Process Next (SPN)
- This algorithm is based on the SJF algorithm, from the point of view of an interactive system.
- The algorithm makes an estimation over the execution time needed in the next quantum, and choose the process with the lowest value.
- This estimation can be improved by using a balanced average value: if T_0 and T_1 are the estimated and real values, the next estimation will be:

  $a \times T_0 + (1-a) \times T_1$

  This is the aging technique, usually based on a=1/2.

PROCESS SCHEDULING INTERACTIVE. ALGORITHMS (6)

- ## Guaranteed scheduling
- The GS algorithm is able to make a “promission” for the performance of the system:
  - if the compromise is made for a system with n processes, then the system will try to offer the n-th part of its resources to each process.
- In order to reach this target it is necessary a good accounting for every available resources.
- The chosen process is the one with the smallest system usage in the last (unit of) time.

PROCESS SCHEDULING INTERACTIVE. ALGORITHMS (7)

- Lottery Scheduling
- This algorithm offers to each process a lottery ticket.
  - Each decision is made up of a simple extraction, the prize being “only” a slice of processor’s time.
- A lot of adjacent activities are allowed:
  - it is possible for a process to hold several tickets (this improves the overall chances for this process).
  - Ticket transactions are allowed.
    - For example, a group of client processes could cooperate such that the chances of the server process they are talking with to be higher when a critical activity is supposed to happen.

OPERATING SYSTEMS

Lecture #6

PROCESS SCHEDULING

Scheduling Algorithms. Batch.

Exercises and demonstrations …

PROCESS SCHEDULING

Interactive Algorithms. Batch.

Exercises and demonstrations