Brief introduction to <u>OS Dispatcher</u>

Presented by:
Amir Froghi
Farshid Noshi
Amirhossein Najafizadeh

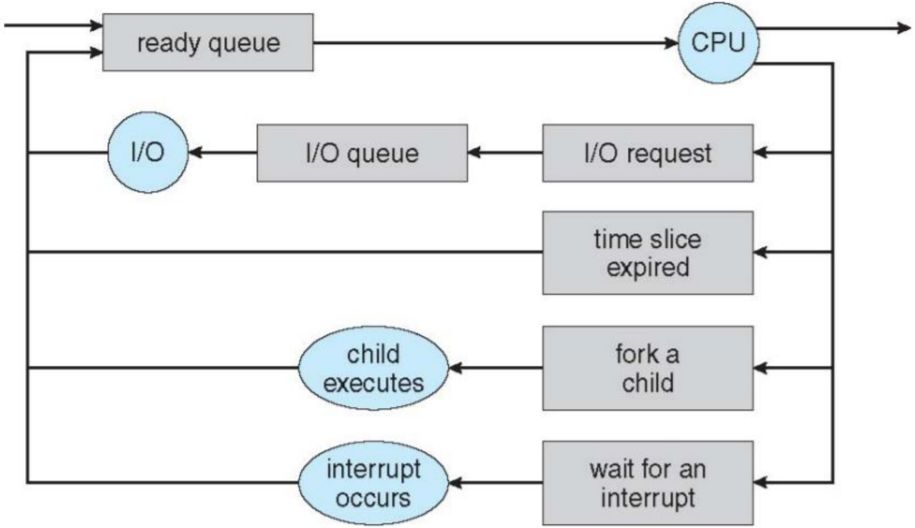
Winter 2021

Subjects

- What is dispatcher ?
  - Quick look at scheduler
- Why does os need dispatcher ?
- How does a dispatcher work?
- Scheduler v.s.Dispatcher
- Dispatching algorithms
- Dispatching types
- Example
- Summery

What is dispatcher ?

Process scheduler selects among available processes for next execution on CPU.



What is dispatcher ???

A dispatcher is a special program which comes into play after the scheduler.
When the scheduler completes its job of selecting a process, it is the dispatcher which takes that process to the desired state/queue.
The dispatcher is the module that gives a process control over the CPU after it has been selected by the short-term scheduler.

Dispatcher functionalities

This function involves the following:

- Switching context
- Switching to user mode
- Jumping to the proper location in the user program to restart that program

A running process goes to the waiting state for IO operation etc., and then the CPU is allocated to some other process. This switching of CPU from one process to the other is called **context switching**.

Why does OS need dispatcher?

When the processes are in the ready state, then the CPU applies some process scheduling algorithm and choose one process from a list of processes that will be executed at a particular instant of time. This is done by a scheduler i.e. selecting one process from a number of processes is done by a scheduler.

Now, the selected process has to be transferred from the current state to the desired or scheduled state. So, it is the duty of the dispatcher to dispatch or transfer a process from one state to another. A dispatcher is responsible for context switching and switching to user mode.

More responsibilities

Sometimes, the Dispatcher is considered part of the short-term scheduler, so the whole unit is called the ***short-term scheduler***.

In this scenario, the task of the short term scheduler is to select a process from the ready queue and allocate the CPU for that process.

In the operating system, a dispatcher has the following responsibilities:

Switching to user mode

All of the low-level operating system processes run on the kernel level security access, but all application code and user issued processes run in the application space or the user permission mode. The Dispatcher switches the processes to the user mode.

Addressing

The program counter (PC) register points towards the next process to be executed. The Dispatcher is responsible for addressing that address.

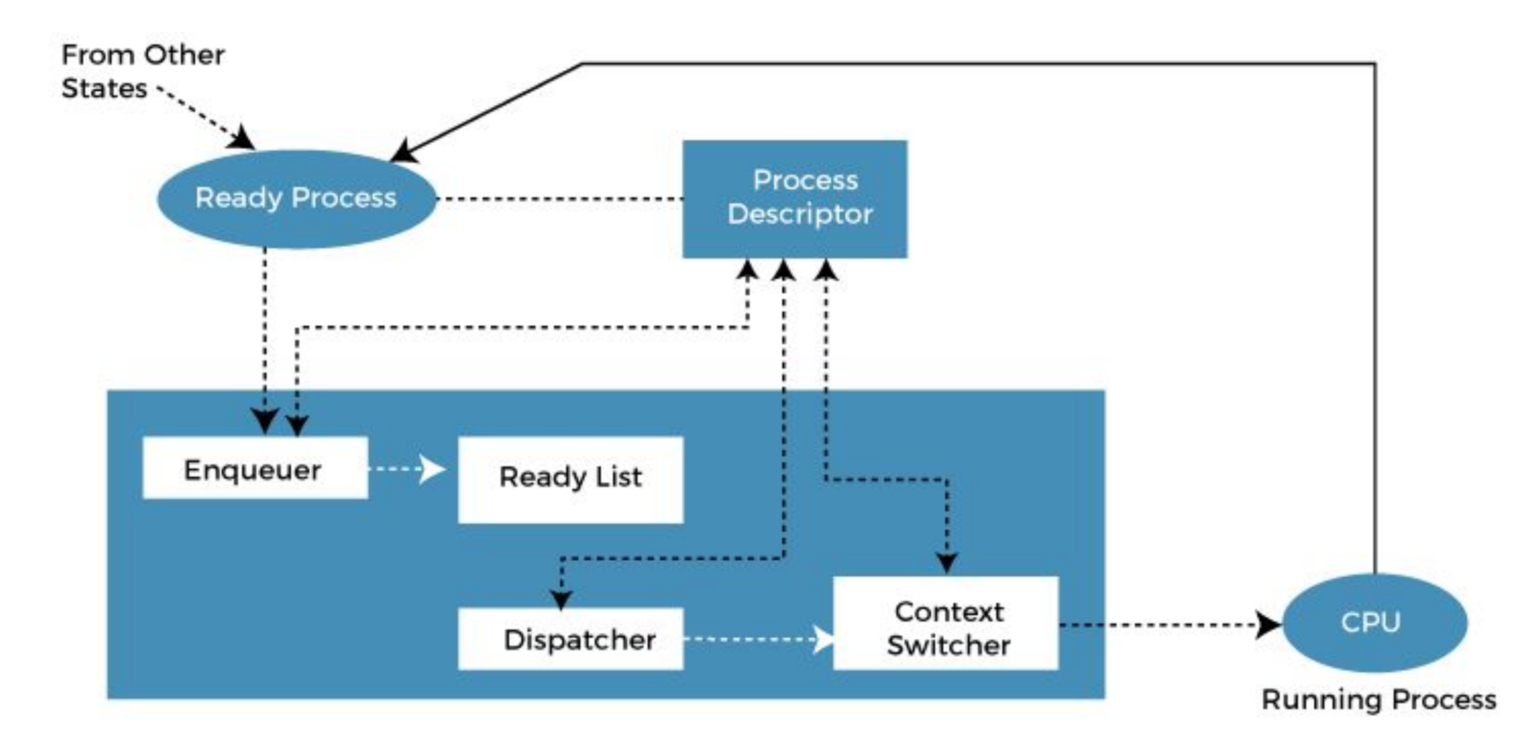
Initiation of context switch

A context switch is when a currently running process is halted, and all of its data and its process control block (PCB) are stored in the main memory, and another process is loaded in its place for execution.

Managing dispatch latency

The Dispatcher needs to be as fast as possible, as it is run on every context switch. The time consumed by the Dispatcher is known as ***dispatch latency***.

Dispatch latency is calculated as the time it takes to stop one process and start another. The lower the dispatch latency, the more efficient the software for the same hardware configuration.


From Other States
Ready Process
Process Descriptor
Enqueuer
Ready List
Dispatcher
Context Switcher
CPU
Running Process

Dispatcher v.s Scheduler

The CPU cannot execute all processes residing in the ready queue and waiting for execution simultaneously. So the operating system has to choose a particular process based on the scheduling algorithm, and the scheduler does this procedure of selecting a process among various processes.

Once the scheduler has selected a process from the queue, the Dispatcher takes it from the ready queue and moves it into the running state. Therefore, the scheduler gives the Dispatcher an ordered list of processes which the Dispatcher moves to the CPU over time.

Dispatcher v.s Scheduler

Scheduler and Dispatcher are used in the process scheduling of an operating system, and they both complete the same process or task.
Still, the difference between scheduler and Dispatcher is that the scheduler selects a process out of several processes to be executed.
In contrast, the Dispatcher allocates the CPU for the selected process by the scheduler.
There are some more differences between the scheduler and the Dispatcher in the operating system, such as:

- The scheduler selects a process from a list of processes by applying some process scheduling algorithm.
  On the other hand, the dispatcher transfers the process selected by the short-term scheduler from one state to another.

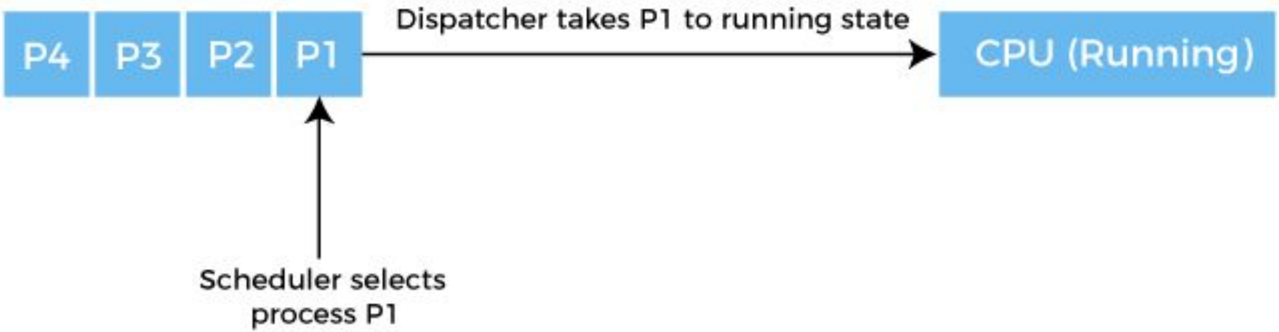
- The scheduler works independently,
  while the dispatcher has to be dependent on the scheduler i.e. the dispatcher transfers only those processes that are selected by the scheduler.

- For selecting a process, the scheduler uses some process scheduling algorithm like FCFS, Round-Robin, SJF, etc.
  But the dispatcher doesn't use any kind of scheduling algorithms.

- The only duty of a scheduler is to select a process from a list of processes.
  But apart from transferring a process from one state to another, the dispatcher can also be used for switching to user mode.
  Also, the dispatcher can be used to jump to a proper location when the process is restarted.

Example of how dispatcher works.

For example, if we have three processes P1, P2, and P3 in the ready state. The arrival time of all these processes is T0, T1, and T2 respectively.


P4
P3
P2
P1
Dispatcher takes P1 to running state
CPU (Running)
Scheduler selects process P1

The process P1 arrives first, so the scheduler will decide it is the first process to be executed, and the Dispatcher will remove P1 from the ready queue and give it to the CPU.

Then the scheduler will determine process P2 to be the next process that should be executed, so when the Dispatcher returns to the queue for a new process, it will take process P2 and give it to the CPU. This continues in the same way for process P3 and then P4.

If we are using the First Come First Serve approach, then the scheduler will first select the process P1 and the dispatcher will transfer the process P1 from the ready state to the running state.

After completion of the execution of the process P1, the scheduler will then select the process P2 and the dispatcher will transfer the process P2 from ready to running state and so on.

How dispatcher functions in modern operating systems

dispatcher is used as an interrupt in the operating system
not all tasks are invoked with a dispatcher clock
because some tasks need more to execute

3 types of dispatcher:
short time
medium time
long time
dispatcher is activated by the system clock which generates an interrupt between 10ms and one per 10 microseconds
multilevel scheduler may cause process starvation

/////next slide///////
the fact that one task works with long time dispatcher and preempts processes working with short time dispatcher does not mean that it has more priority