CHAPTER 3

THE PROCESS CONCEPT

Dr. Soha S. Zaghloul

LAYOUT

- Process Definition
- Process State
- Process Control Block

PROCESS DEFINITION (1)

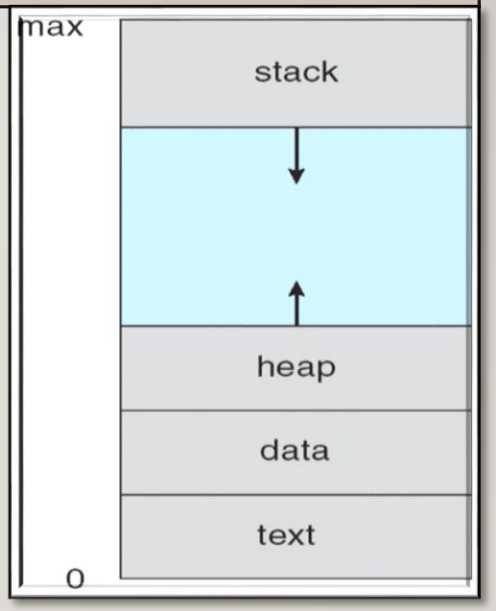
A process is a program in execution. This is also called a job.

A program is a passive entity; whereas the process is an active one.

In other words, a program becomes a process when it is loaded into memory.

A process includes all the information necessary to make it run during its lifetime such as:

- The program code, also known as the text section.
- The program counter that includes the address of the next instruction to be executed.
- The final contents of the processor's registers during the execution of the process.
- The stack: this contains the functions (methods) parameters, return addresses, and global variables.
- The heap: this contains the dynamically allocated memory required during run time.



PROCESS DEFINITION (2)

Although two processes may be associated with the same program code (text section), they are nevertheless considered as two separate execution sequences.

The following are some illustrating examples:

- Several users may be running different copies (processes) of the mail program.
- The same user may be running many copies (processes) of the web browser program.

In the above examples, the text sections are equivalent; but the data, heap, and stack versions vary.

A process itself may be an execution environment for other code:

As an example:

- The Java Virtual Machine (JVM) runs as a process.
- However, JVM interprets the loaded code (your program), and takes actions on behalf of that code.

PROCESS STATE (1) – INTRODUCTION

The current activity of a process is known as the process state.

This means that the process state continuously changes with time during a process lifetime.

A process may be in one of the following states during its lifetime:

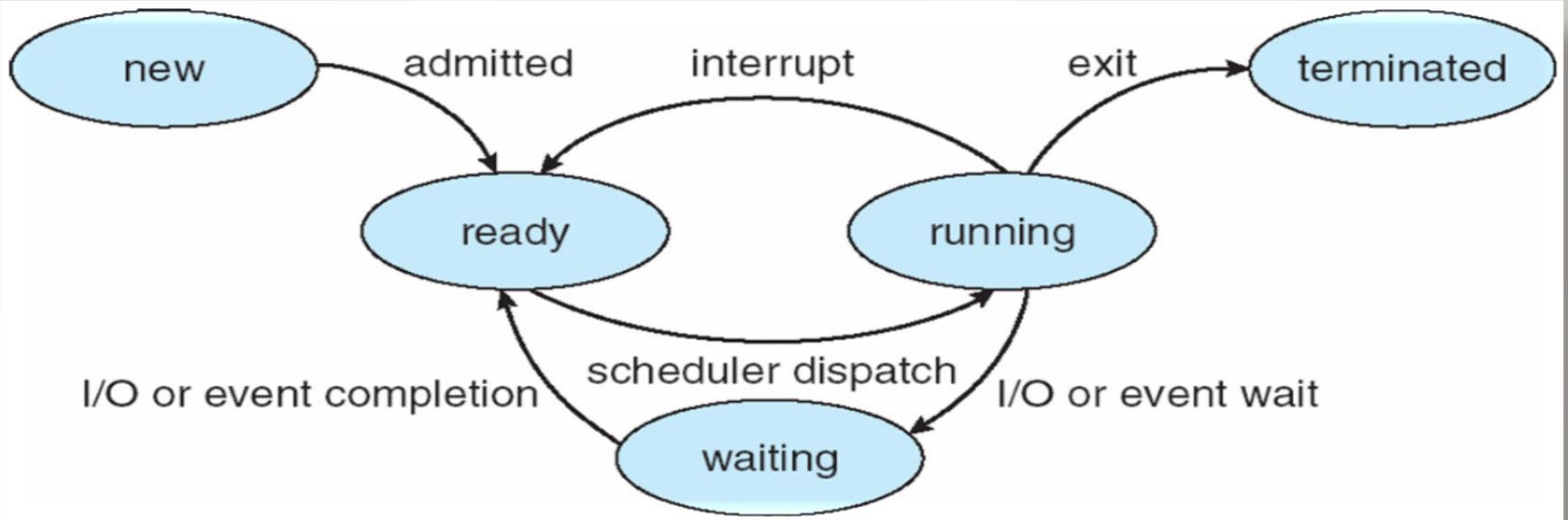
- New: the process is being created.
- Running: the process instructions are being executed.
- Waiting: the process is waiting for some event to occur. For example, for the value of an input variable.
- Ready: the process ready for execution but is waiting to be assigned to a processor.
- Terminated: the process has finished execution.

A processor executes at most a single instruction at any instant of time.

The instructions executed by a processor during a specific period of time might belong to one or more processes.

Note that many processes might be ready at an instant of time. In such case, they contend (compete) for the processor(s).

PROCESS STATE (2) – THE STATE DIAGRAM (1)

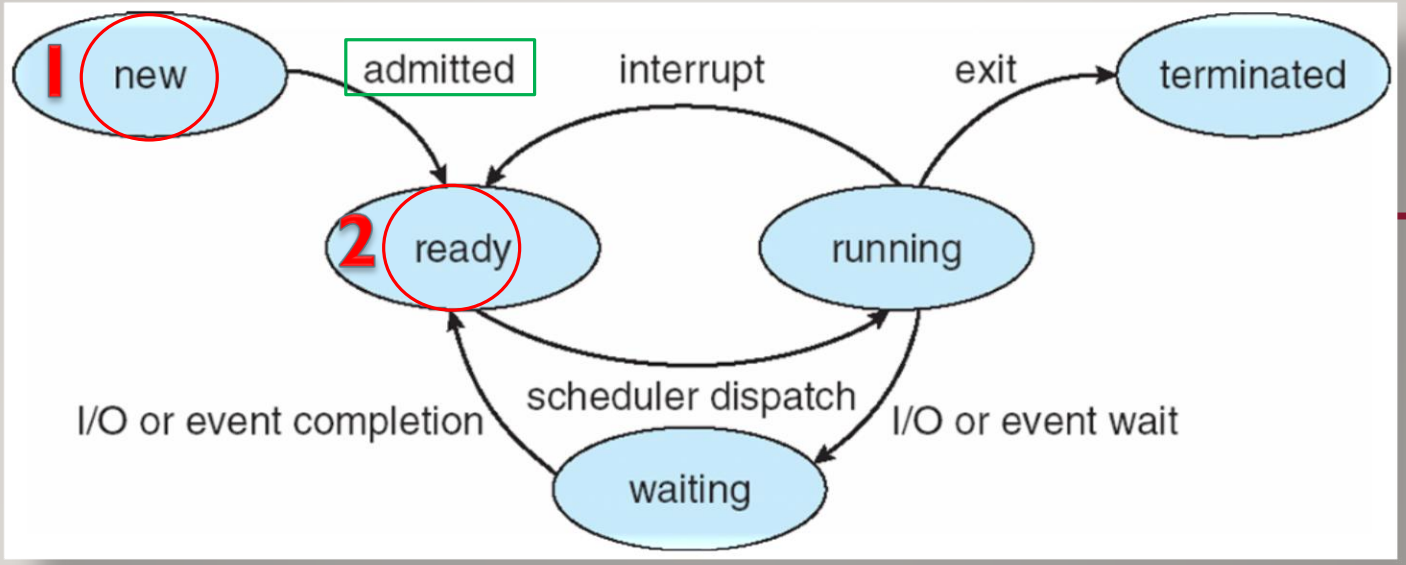


A state diagram consists of states and actions.

The process state diagram depicts the following:

- All possible states that a process can go through during its lifetime.
- States are shown in circles.
- All possible actions that cause a process to change its state.
- Actions are shown with arrows.
- A transition consists of two states: the previous state and the current one, plus the action that caused such state change.
- For example, consider the following transition: a process in the running state changes to the ready state when interrupted.

PROCESS STATE (3) – THE STATE DIAGRAM (2) – THE NEW & READY STATES



A written program is initially on the disk in the job pool.

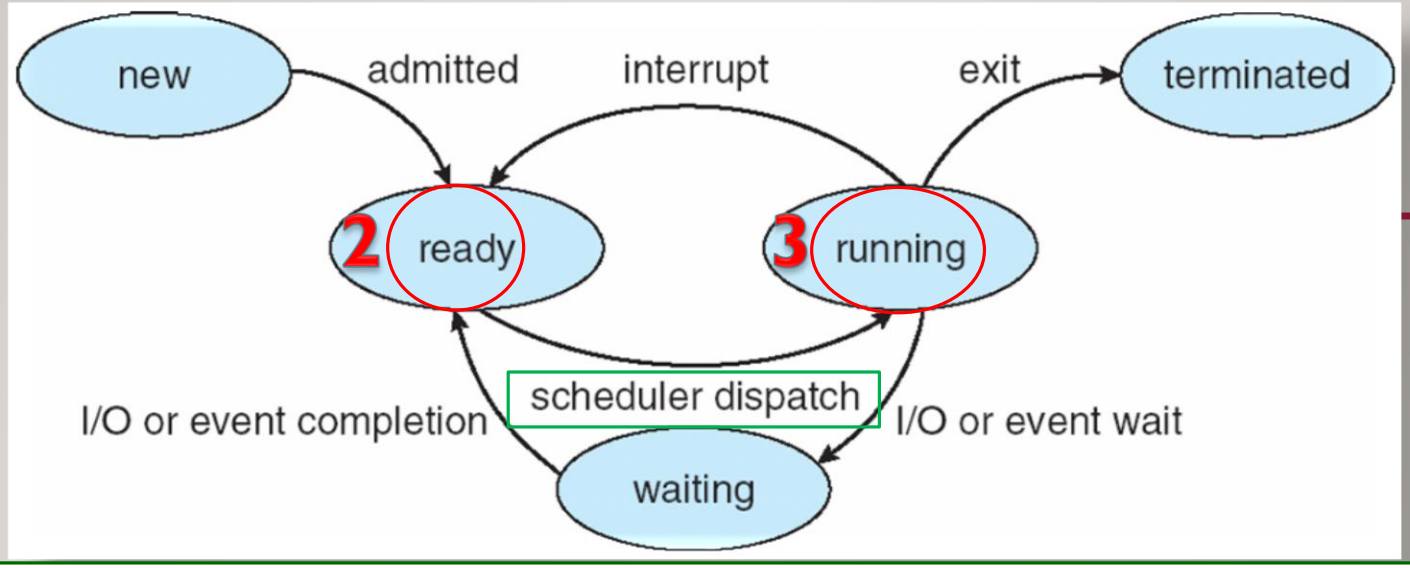
When the OS selects the program to be loaded into memory, a process is created.

At this instant of time, the process is given the status new.

When the OS creates a new process, it assigns to it a unique identifier (number). This is known as the process identifier.

The process is soon admitted and it becomes ready for execution.

PROCESS STATE (4) – THE STATE DIAGRAM (3) – THE READY & RUNNING STATES



Therefore, a process in the ready state resides in the memory. We say that the process is in the ready queue. It waits the OS to assign him a processor to execute on.

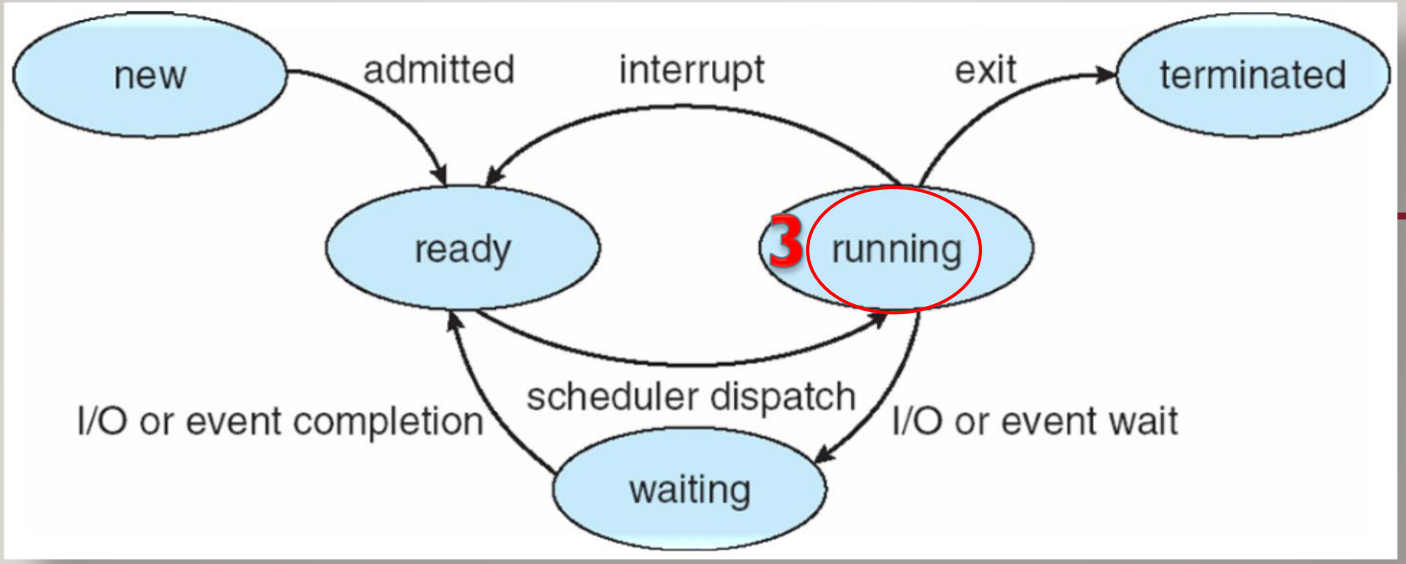
Note that we may have multiple processes in the ready queue competing for the processor.

The OS selects only one process to be assigned the processor. Selection is made by a specific part of the OS called the CPU scheduler.

The CPU scheduler is made to select a process according to specific criteria.

As soon as the process is assigned a processor, it executes and changes to the Running state. We say that the scheduler dispatches the process.

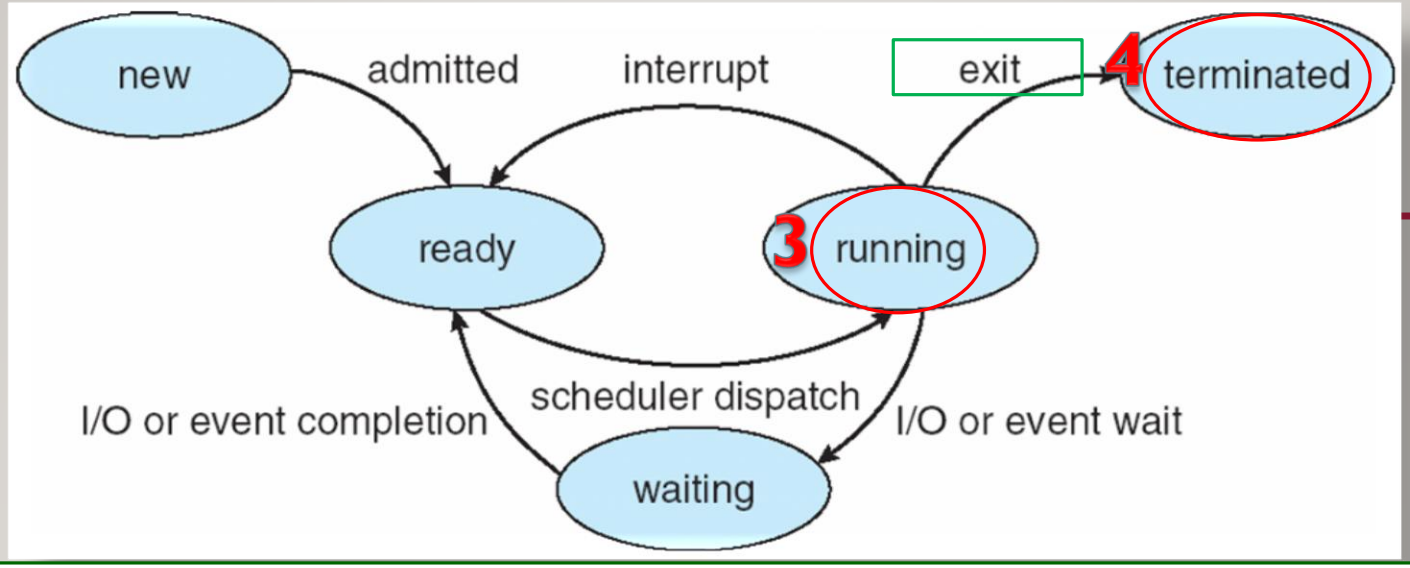
PROCESS STATE (5) – THE STATE DIAGRAM (4) – THE RUNNING STATE



While running, a process may change its state to one of the following:

- Terminate
- Wait
- Ready (again…)

PROCESS STATE (6) – THE STATE DIAGRAM (5) – THE RUNNING & TERMINATED STATES

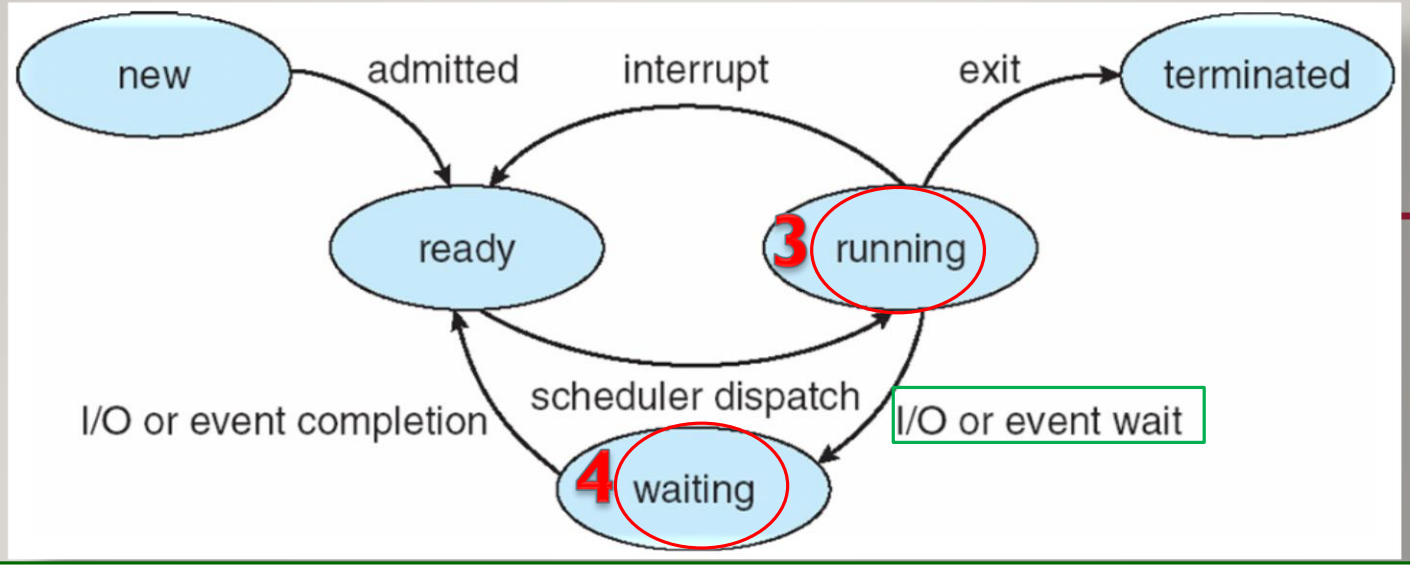


A process may run on the processor until it completes execution.

In this case, the process exits, changes to the terminated state.

A terminated process deallocates (leaves) the memory, and all previously allocated resources.

PROCESS STATE (7) – THE STATE DIAGRAM (6) – THE RUNNING & WAITING STATES



While running a process, a processor may encounter an I/O statement.

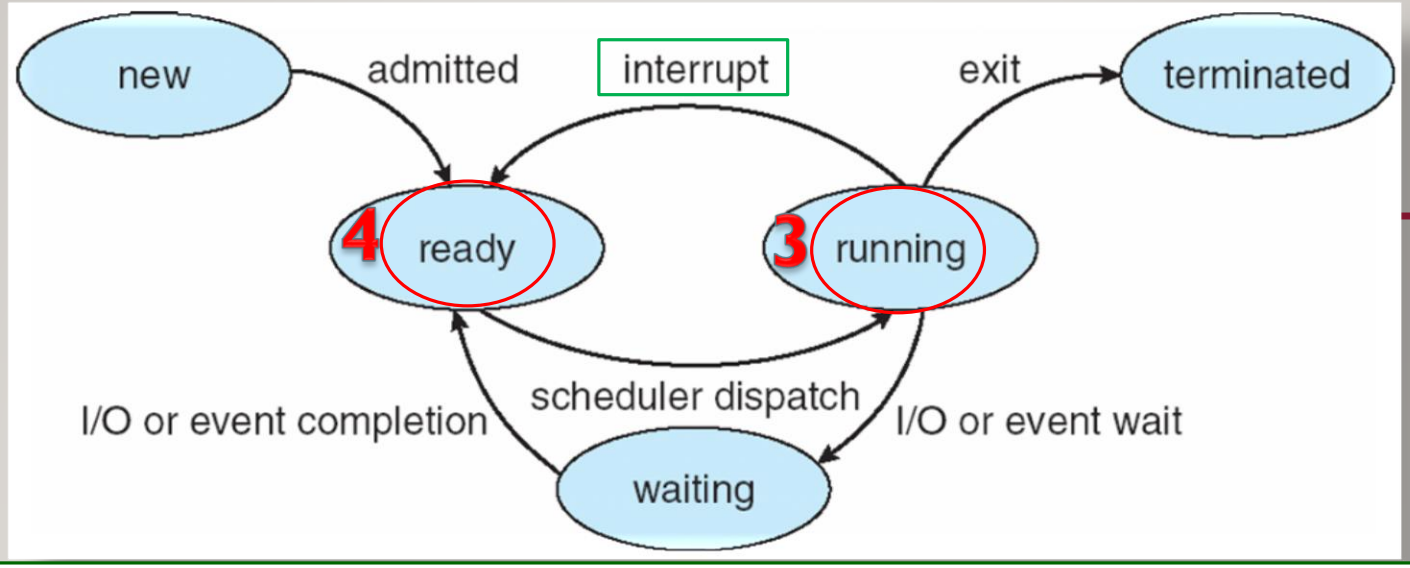
Assume for example, that the processor encountered an "input" statement.

In this case, the process halts execution until the user enters the input. The processor speed is millions times faster than pressing a single key on the keyboard.

In order to make full use of the processor, the process is appended to a device waiting queue. The process status then changes to the waiting state.

During the time the process is in the device waiting queue, the processor runs another ready process in the ready queue, if any. Otherwise, it stays idle.

PROCESS STATE (7) – THE STATE DIAGRAM (6) – THE RUNNING & READY STATES



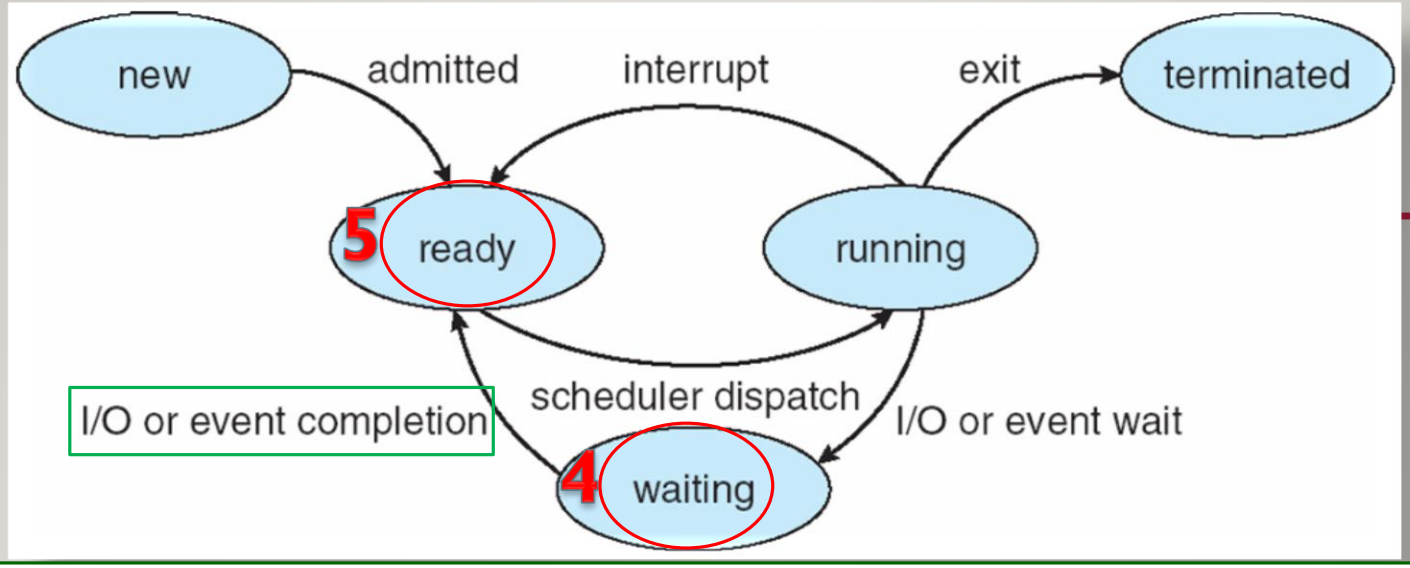
In a multi-process environment, the OS gives a quantum time to each running process.

When the process exceeds its quantum time, the OS interrupts it and returns it back to the ready queue.

In other words, its status changes from running to ready.

The process then waits to be selected by the OS to be re-assigned the CPU.

PROCESS STATE (8) – THE STATE DIAGRAM (7) – THE WAITING & READY STATES



A process waits for the I/O device in the device waiting queue. As soon as it is assigned the device, it performs the I/O operation.

When the process completes the I/O operation, for example, gets the value of the input variable from the user, it becomes ready to resume execution.

Therefore, the status changes from wait to ready.

The process then waits in the ready queue until it is assigned a CPU by the OS.

PROCESS STATE (9) – NOTES

The state diagram proves that a CPU does not dedicate itself to a processor from the start of its execution till the end.

In fact, an efficient OS aims to make full use of the available resources.

The CPU scheduler guarantees that processes complete execution in an optimum time. This will be studied in details later in this course.

As the number of the processes in the ready queue increases, as there is a less probability that the CPU stays idle.

The state diagram also explains how a CPU executes a single instruction at a time, however, instructions may belong to multiple processes.

In a multiple CPU environment, two or more processes may be dispatched from the ready queue in the same cycle to different processors.

PROCESS CONTROL BLOCK

Each process is represented to the OS by its Process Control Block (PCB).

A PCB contains the following information about the corresponding process:

- The process state: New/Ready/Running/Waiting/Terminated.
- The process identifier: The OS assigns a unique number to each process as soon as it is created.
- The program counter: This holds the address of the next instruction to be executed.
- The CPU registers: The final values produced by the process at an instant of time and stored in the CPU registers. When a process is interrupted by an ISR for example, these values are stored in the PCB. When the process resumes execution, they are restored. We say that the process state is saved/restored.
- Memory Management Information: The values of the base and limit registers, and the page tables. This are to be explained later in this course.
- I/O Status Information: This includes the list of I/O devices allocated to the process & list of open files.
- CPU Scheduling Information: This includes the process priority & pointers to scheduling queues.
- Accounting Information used for billing. It includes the amount used of CPU, A/C numbers, etc…