Process

What is a Process?

- A process is a program in execution.
- Process is not as same as program code but a lot more than it.
- A process is an 'active' entity as opposed to program which is considered to be a 'passive' entity.
- Attributes held by process include hardware state, memory, CPU etc.

Process memory

Process memory is divided into four sections for efficient working :

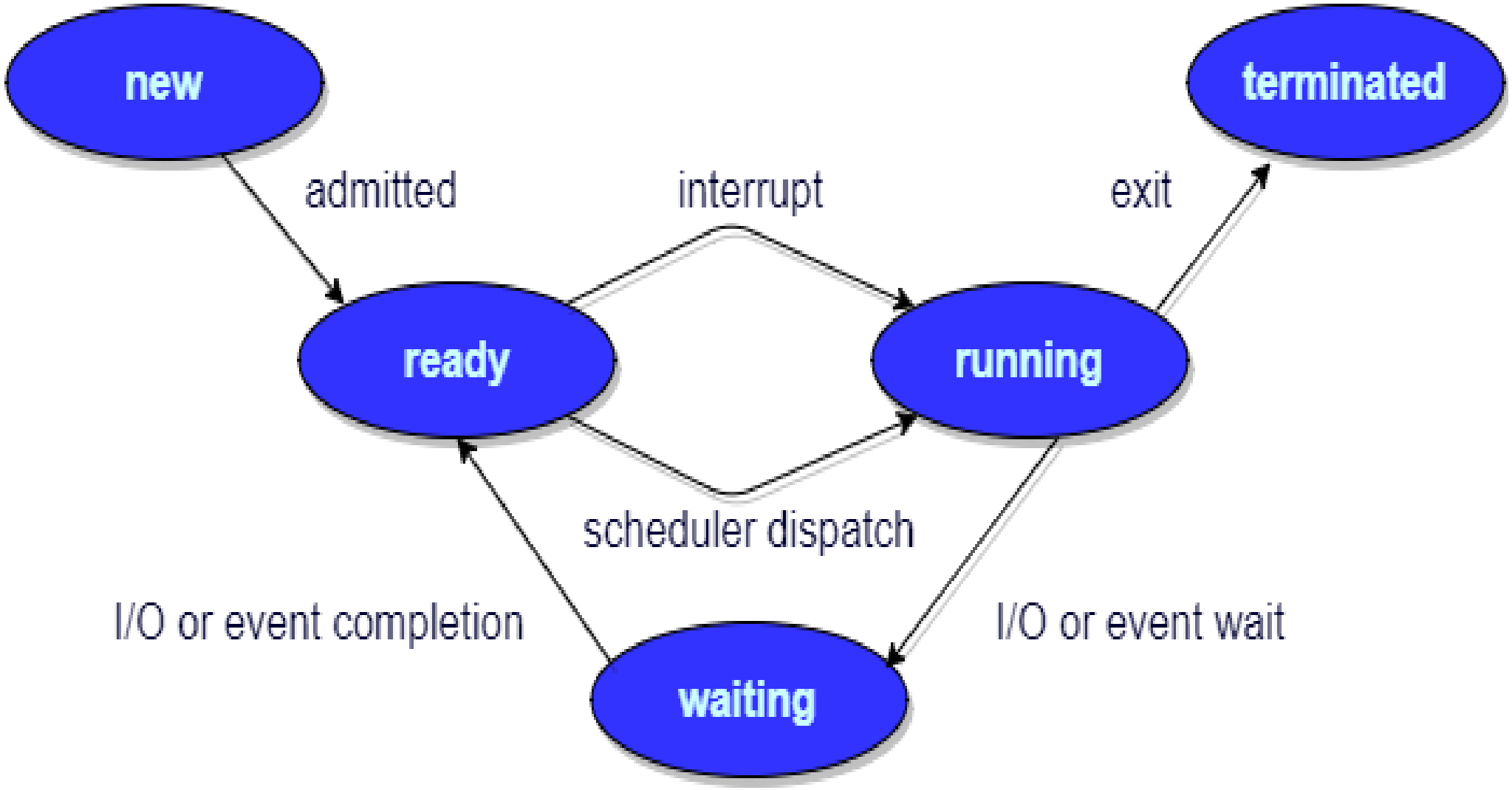
- **Text section** is made up of the compiled program code, read in from non-volatile storage when the program is launched.
- **Data section** is made up the global and static variables, allocated and initialized prior to executing the main.
- **Heap** is used for the dynamic memory allocation, and is managed via calls to new, delete, free, etc.
- **Stack** is used for local variables. Space on the stack is reserved for local variables when they are declared.

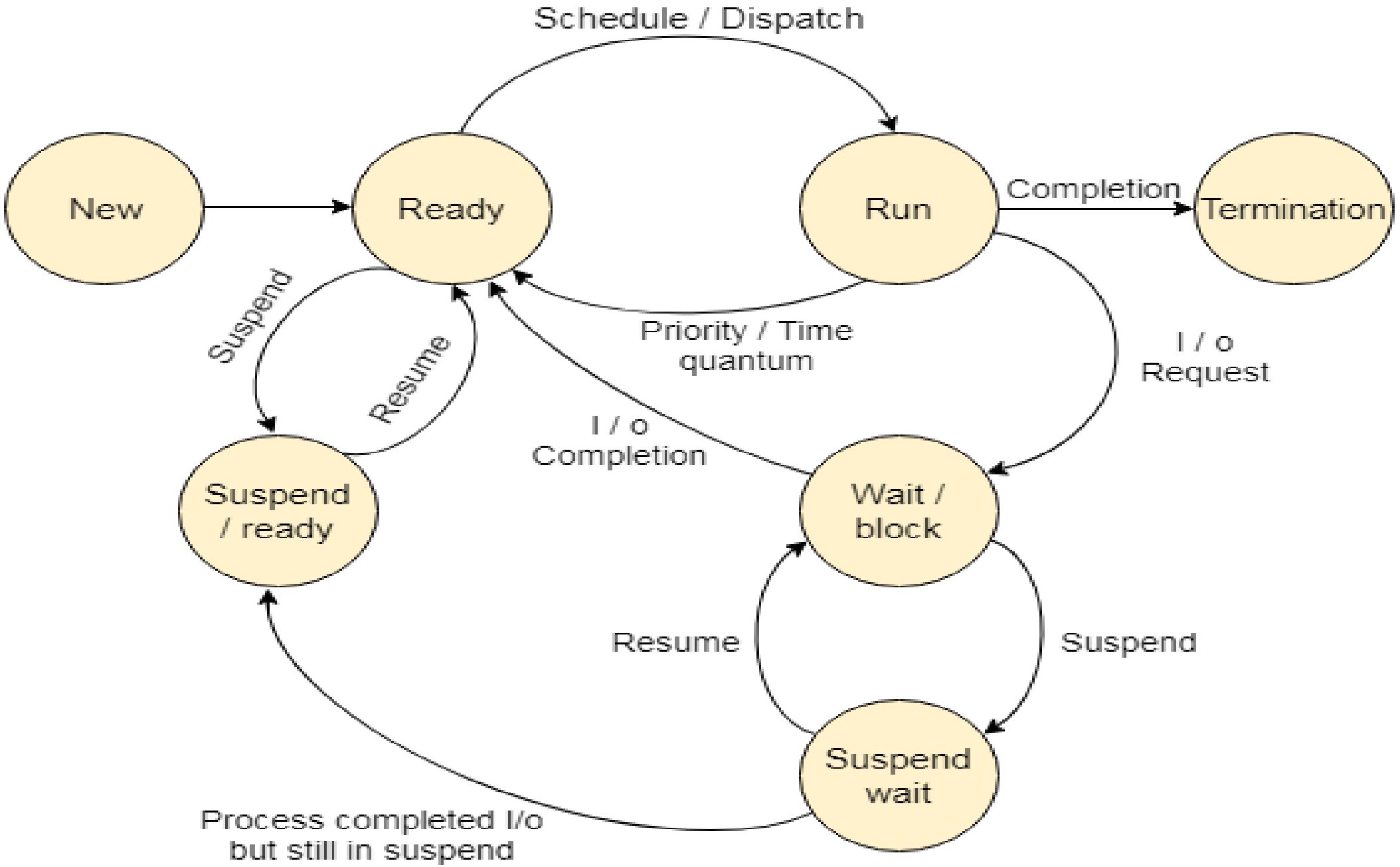
Different Process States

Processes in the operating system can be in any of the following states:

- NEW- The process is being created.
- READY- The process is waiting to be assigned to a processor.
- RUNNING- Instructions are being executed.
- WAITING- The process is waiting for some event to occur(such as an I/O completion or reception of a signal).
- TERMINATED- The process has finished execution

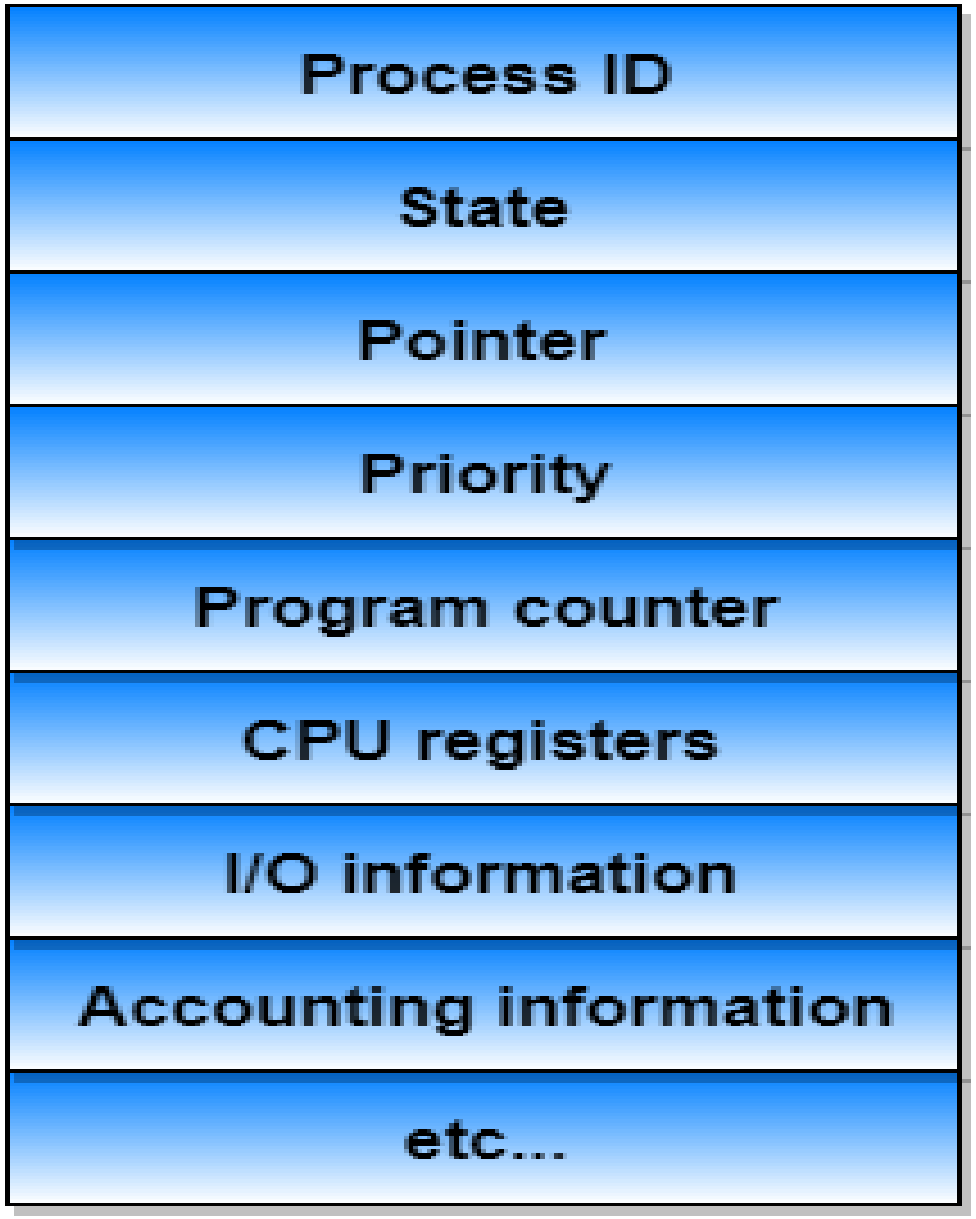
Different Process States




Schedule / Dispatch
New
Ready
Run
Completion
Termination
Suspend
Resume
Priority / Time quantum
I / o Request
I / o Completion
Suspend / ready
Wait / block
Resume
Suspend
Suspend wait
Process completed I/o but still in suspend

Process Control Block

- There is a Process Control Block for each process, enclosing all the information about the process. It is a data structure, which contains the following:
- **Process State**: It can be running, waiting etc.
- **Process ID** and the **parent process ID**.
- CPU registers and Program Counter. **Program Counter** holds the address of the next instruction to be executed for that process.
- **CPU Scheduling** information: Such as priority information and pointers to scheduling queues.
- **Memory Management information**: For example, page tables or segment tables.
- **Accounting information**: The User and kernel CPU time consumed, account numbers, limits, etc.
- **I/O Status information**: Devices allocated, open file tables, etc.


Process ID
State
Pointer
Priority
Program counter
CPU registers
I/O information
Accounting information
etc...

What is Process Scheduling?

- The act of determining which process is in the **ready** state, and should be moved to the **running** state is known as **Process Scheduling**.
- The prime aim of the process scheduling system is to keep the CPU busy all the time and to deliver minimum response time for all programs.
- For achieving this, the scheduler must apply appropriate rules for swapping processes IN and OUT of CPU.

Scheduling fell into one of the two general categories:

- **Non Pre-emptive Scheduling:** When the currently executing process gives up the CPU voluntarily.
- **Pre-emptive Scheduling:** When the operating system decides to favour another process, pre-empting the currently executing process

What are Scheduling Queues?

- All processes, upon entering into the system, are stored in the **Job Queue**.
- Processes in the Ready state are placed in the **Ready Queue**.
- Processes waiting for a device to become available are placed in **Device Queues**.
- A new process is initially put in the **Ready queue**.
- It waits in the ready queue until it is selected for execution (or dispatched).

Once the process is assigned to the CPU and is executing, one of the following several events can occur:

- The process could issue an I/O request, and then be placed in the **I/O queue**.
- The process could create a new subprocess and wait for its termination.
- The process could be removed forcibly from the CPU, as a result of an interrupt, and be put back in the ready queue.