Process concept

OPERATING SYSYTEM

SAVIYA VARGHESE
Dept of BCA

2020-21

Processes

- Process Concept
- Process Scheduling
- Operation on Processes
- Cooperating Processes
- Interprocess Communication

Introduction

A *process* can be thought of as a program in execution. A process will need certain resources—such as CPU time, memory, files, and I/O devices—to accomplish its task. These resources are allocated to the process either when it is created or while it is executing.

A process is the unit of work in most systems. Such a system consists of a collection of processes: Operating-system processes execute system code, and user processes execute user code. All these processes may execute concurrently.

Process Concept

- Process – a program in execution; process execution must progress in sequential fashion.
- 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 :

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

Process State

- As a process executes, it changes *state*
  - new: The process is being created.
  - ready: The process is waiting to be assigned to a process.
  - running: Instructions are being executed
  - waiting: The process is waiting for some event to occur.
  - terminated: The process has finished execution.

New state

- This is the initial state when a process is first started/created.
- A program which is going to be picked up by the OS into the main memory is called a new process.

Ready state

- **Whenever a process is created, it directly enters in the ready state, in which, it waits for the CPU to be assigned.**
- Ready to run. After the creation of a process, the process enters the ready state i.e. the process is loaded into the main memory.
- The process here is ready to run and is waiting to get the CPU time for its execution.
- Processes that are ready for execution by the CPU are maintained in a queue for ready processes.
-

Ready queue

- set of all processes residing in main memory

Dispatching

- The act of assigning a processor to the first process on the ready list is called dispatching , and is performed by a s/m entity called the dispatcher.

- When a process is dispatched, it transitions from *ready* to *running*

RUNNING

- One of the processes from the ready state will be chosen by the OS depending upon the scheduling algorithm
- The process is chosen by CPU for execution and the instructions within the process are executed by any one of the available CPU

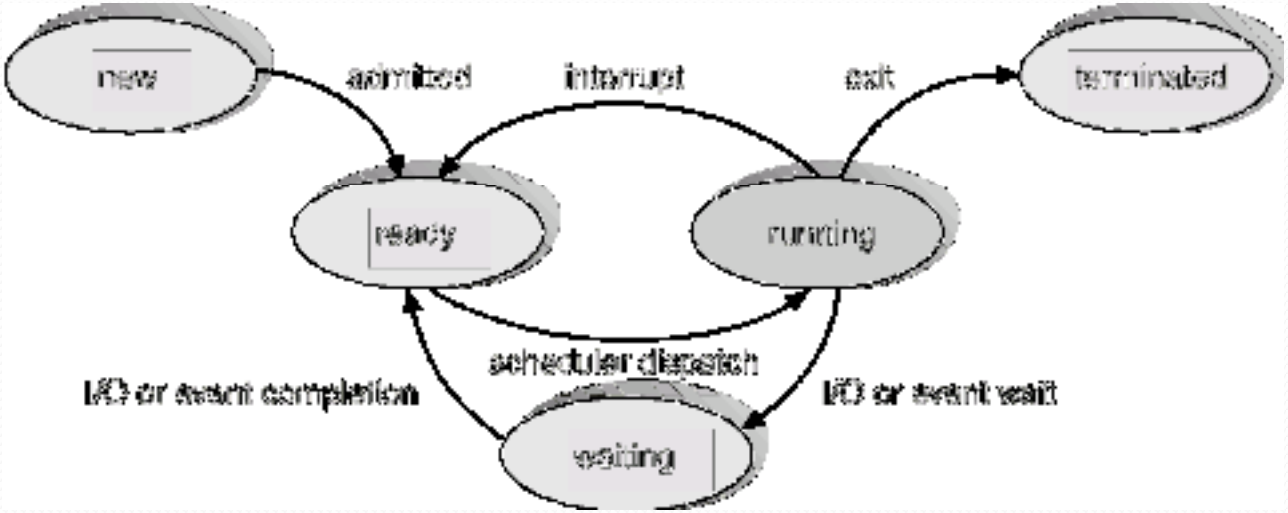
WAITING

- Whenever the process requests access to I/O or needs input from the user or needs access to a critical region(the lock for which is already acquired) it enters the blocked or wait state.
- The process continues to wait in the main memory and does not require CPU.
- Once the I/O operation is completed the process goes to the ready state.

TERMINATED

- When a process finishes its execution, it comes in the termination state.

Diagram of Process State



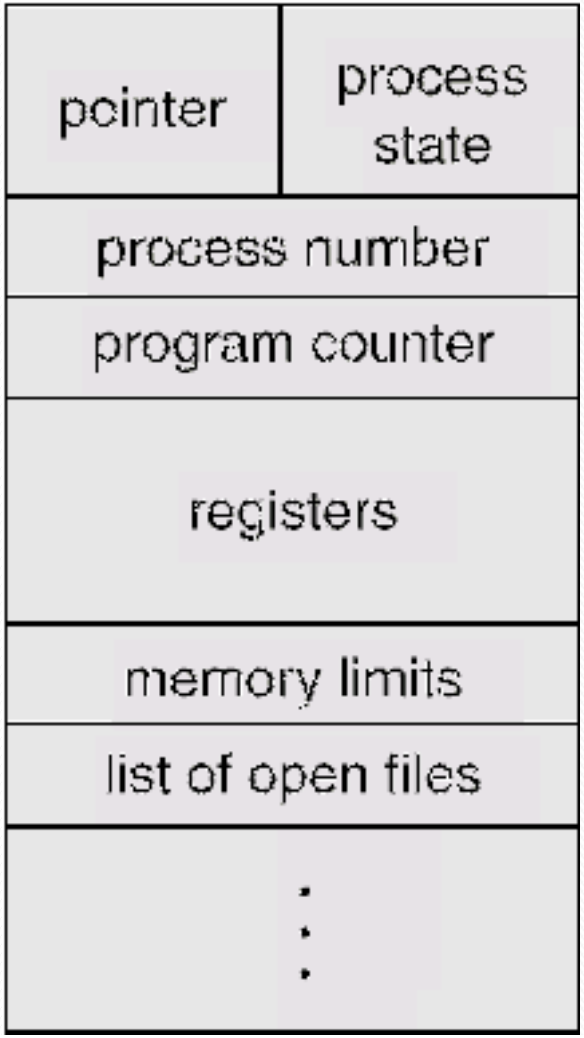
- Process states
  - The act of assigning a processor to the first process on the ready list is called dispatching
  - The OS may use an interval timer to allow a process to run for a specific time interval or quantum
  - Cooperative multitasking lets each process run to completion
- State Transitions
  - At this point, there are four possible state transitions
    - When a process is dispatched, it transitions from *ready* to *running*
    - When the quantum expires, it transitions from *running* to *ready*
    - When a process blocks, it transitions from *running* to *blocked*
    - When the event occurs, it transitions from *blocked* to *ready*

Process Control Block (PCB)

Information associated with each process.

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information

Process Control Block (PCB)



CPU Switch From Process to Process