Five state model

- Improvement of 2-state model.
- ~~scribbled~~
- ~~scribbled~~

New — Processes that are created but not ready

Ready — Admit → **Ready** — dispatch → **Running** — release → **Exit**

timeout (Running → Ready)

Event occurs (Blocked → Ready)

Event wait (Running → Blocked)

Blocked — processes that cannot execute until some event occurs.
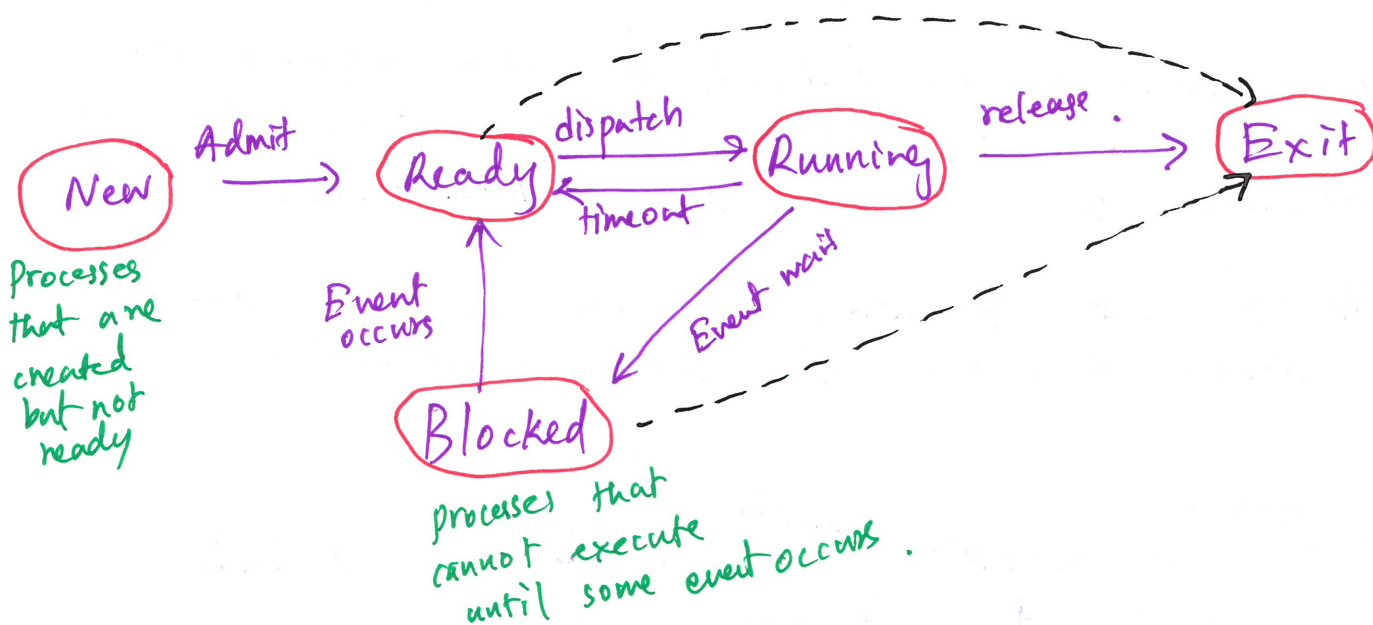
Following is explanation ~~is~~ of few selected transitions from the above diagram.
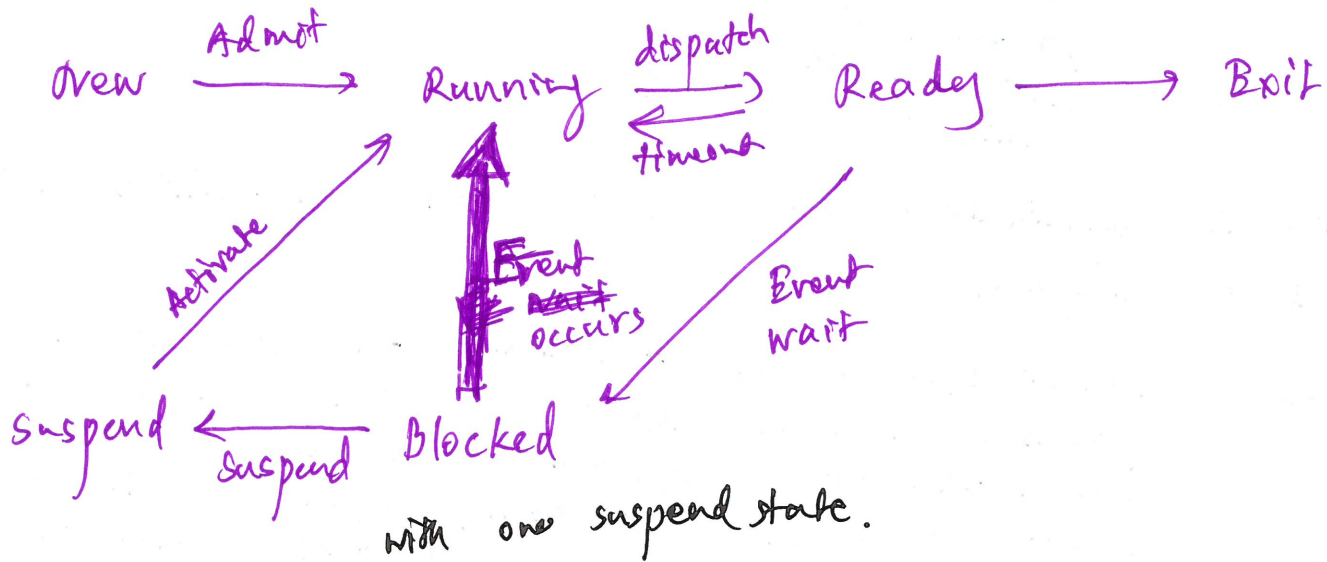
New → Ready : Most system has ~~a~~ a limit on the # of processes that is allowed to run based on the load of the system. OS moves a process from New to READY when OS is ready to ~~dispatch the process.~~ take additional processes.

Running → Ready : Can happen when a process has executed maximum allowed time given in the system. Also this can happen if a new process with higher priority comes in, then the currently running process can be put to ready pool and high priority process will be executed.

1

Ready → Exit
Blocked → Exit : If parent terminates child or parent is termin

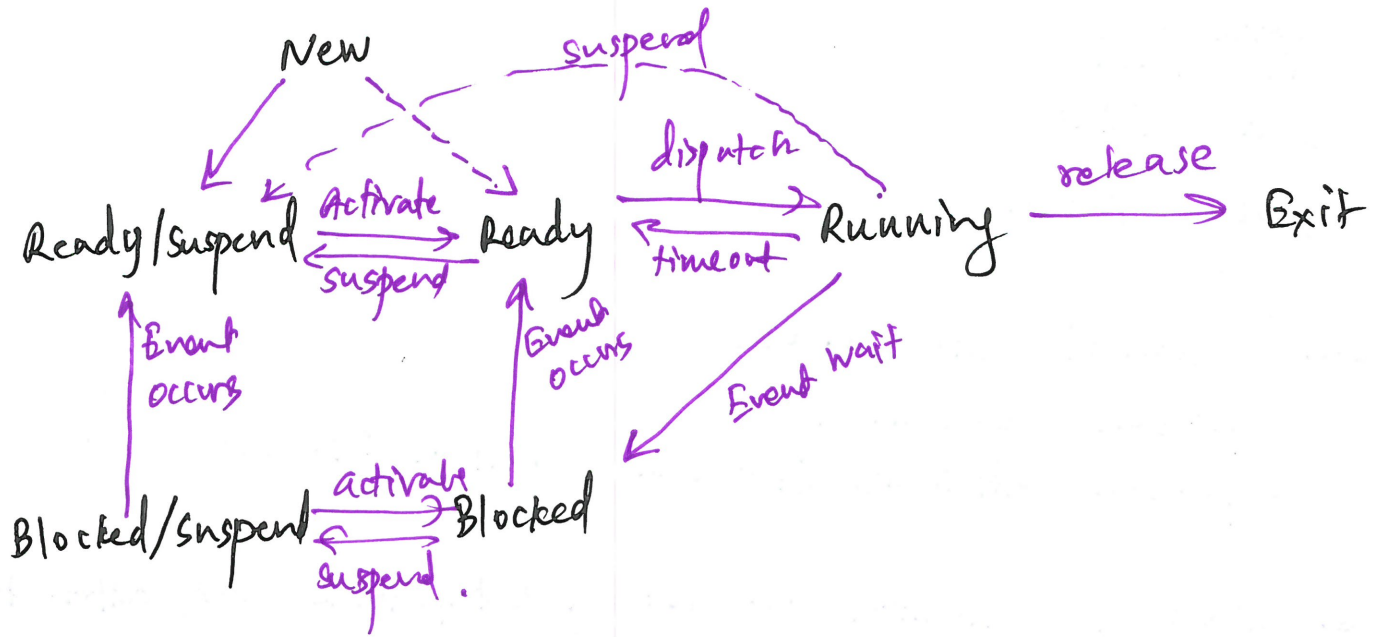## suspending processes.

- Good justifaction for adding more states would be to represent suspended processes.

- A system can be designed to move processes in MM to secondary storage to make space for new processes to come into the MM.

- Following state transition can be used to describe a process when we add the suspend state.

New →(Admit)→ Running ⇄(dispatch / timeout) Ready →→ Exit

Suspend ←(Suspend)← Blocked, with Activate → Running, Event occurs → Running, Event wait → Blocked

with one suspend state.

- However there is a problem with this design, A suspended process which is waiting for an event can be either waiting or ready for execution.

- Therefore, we improve this by further dividing suspend state into: suspend/Ready and suspend/blocked.

Seven state model

New → Ready/Suspend

New → Ready

suspend (New → Running)

Ready/Suspend ⇄ Ready (Activate / suspend)

Ready ⇄ Running (dispatch / timeout)

Running → Exit (release)

Ready/Suspend ← Blocked/Suspend (Event occurs)

Ready ← Blocked (Event occurs)

Running → Blocked (Event wait)

Blocked/Suspend ⇄ Blocked (activate / suspend)

characteristics of a suspended process

- process is not immediately available for execution
- process may or may not be waiting for an event.
- A process was placed in the suspended state by an agent; either itself; a parent process; or the OS.
- The process may not be removed by the suspend state until the agent explicitly orders the removal.
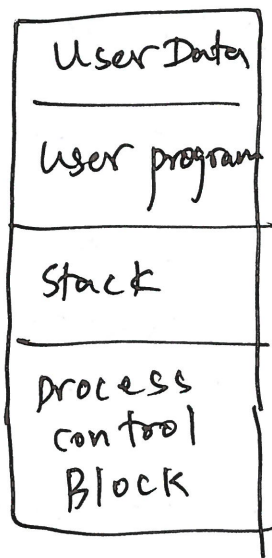
3

OS control structures.

- memory tables
- ~~I/O~~ Device Tables
- Device Tables
- Process Tables.

Process control structures.

- process Location
- process Attributes (Process control Block)

Process Image: collection of program, data, stack, and attributes

User Data
User program
Stack
process control Block

Process Image.

Process Control Block

1. process Identification
2. Processor State Information
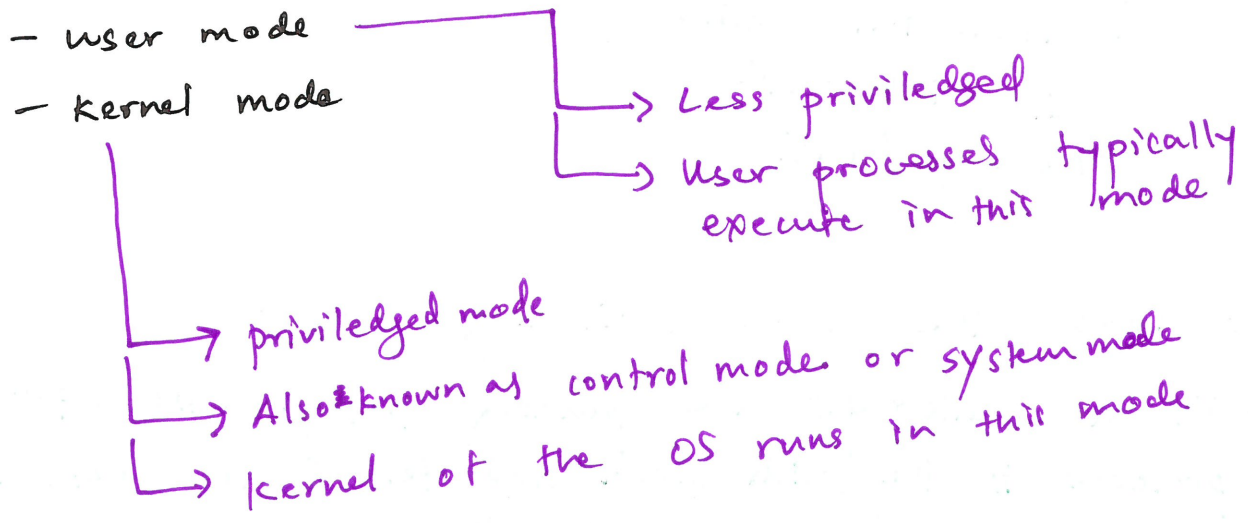3. Process control Information.

4

Interrupt
- clock interrupt.
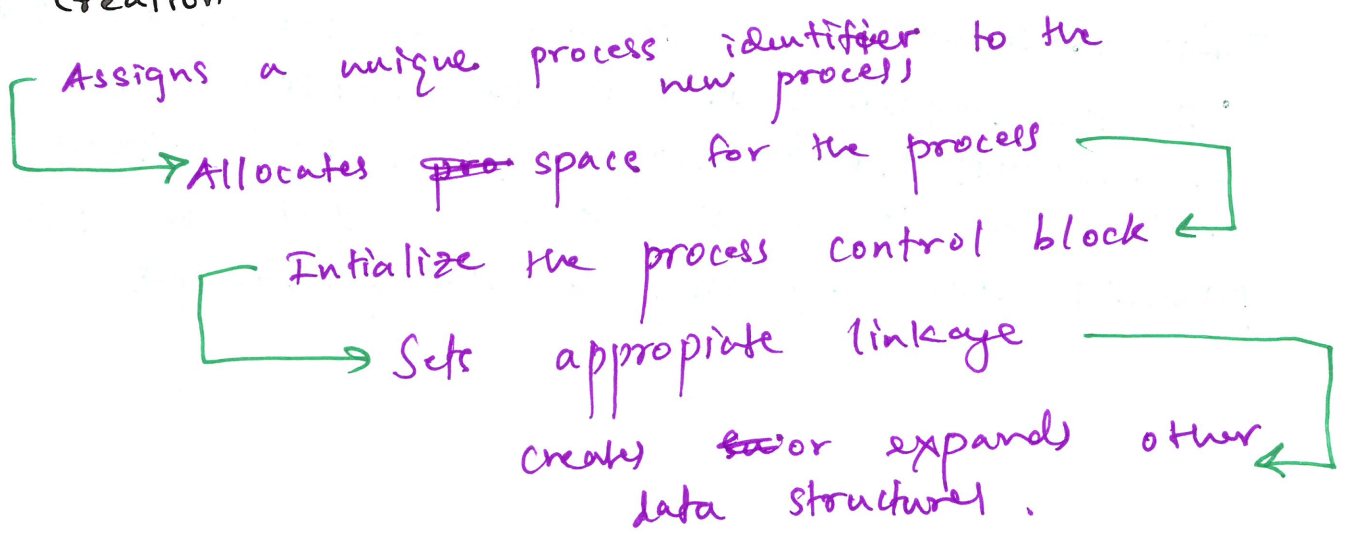- I/O interrupt.
- Memory fault.

How does the mode switching happens?

- If no interrupts are pending the processor; processor proceeds to the fetch stage and fetches the next instruction of the current program in the current process

- If an interrupt is pending the processor; processor sets the program counter to the starting instruction of an interrupt handler program, then it switches from user mode to kernel mode so that the interrupt processing code may run in the priviledged setting.

Modes of execution

- user mode
- kernel mode

→ Less priviledged
→ User processes typically execute in this mode

→ priviledged mode
→ Also known as control mode or system mode
→ kernel of the OS runs in this mode

Process Creation

Assigns a unique process identifier to the new process

→ Allocates space for the process

Intialize the process control block

→ Sets appropiate linkage

Creates or expands other data structures.

How can we interrupt a process

→ Interrupt	External to the execution of the current instruction	Reaction to an external event (Event occurs)
→ Trap	Associated with the execution of the current instruction	Handling an error or an exception handling
→ supervisor call/system call	Explicit request	call to an operating system function.

6