(ISLM-225)

Introduction to Operating System

Department of Information Science & Library Management (ISLM)
University of Rajshahi, Rajshahi-6205, Bangladesh

Unit-1: Operating System Overview: Introduction to OS. Operating system functions, evaluation of O.S., Different types of O.S.: batch, multi-programmed, time-sharing, real-time, distributed, parallel.

Unit-2: System Structure: Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, operating system structure (simple, layered, virtual machine), O/S services, system calls.

Unit-3: Installing and Configuring OS: Introduction to Installation and Media Types, Performing a Custom OS Installation, Run Levels and the Startup/Shutdown Sequence, Logging In and Out of a Operating System.

Unit-4: Process Management: Processes- Concept of processes, process scheduling, operations on processes, co-operating processes, interprocess communication, Threads- overview, benefits of threads, user and kernel threads., CPU scheduling, process synchronization, deadlocks- system model, deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

Unit-6: Storage Management: Memory Management- background, logical vs. physical address space, swapping, contiguous memory allocation, paging, segmentation, segmentation with paging, Virtual Memory- background, demand paging, performance, page replacement, page replacement algorithms (FCFS, LRU), allocation of frames, thrashing, File Systems, I/O Management, Disk Management.

Unit-7: Distributed OS and File System: Motivation, Types of Network-based OS, Network structure, Distributed File System- Background, Naming and transparency, Remote File Access, State full and Stateless services. Distributed Synchronization: Event Ordering, Mutual Exclusion, Atomicity, Concurrency Control, Deadlock Handling, Election algorithm and Reaching agreement.

Process Management

One of the most important part of operating system

Processes

- First of all before we understand processes, lets try to understand how a program is developed.
- So, whenever you want to make a program, what you do is, you first write the program in some high level languages.
- For example, lets say you write the program in C, C++, Java or something like that.
- And that program is written in a higher language.
- Now, we know at the basic level, computer does not understand the high level languages. But it understand only the binary codes, which are 0’s and 1’s.

Processes

- So, the program has to be converted to binary code.
- For that using a compiler, you compile your program and it helps to convert that program into machine code which is understandable by your machine.
- So, after that you have your program which is converted into binary executable code and it is ready for execution.
- But it is not enough to just have that binary code for a program to execute or for a program to tell the computer what it wants to do.

Processes

- So, what it has to do is, it has to be loaded into the memory.
- And for a program to execute, it needs some resources of the computer system.
- So, who will allocate the resources?
- What is the brain that works behind all these?
- It is the OS, which we has been discussing so far.
- So, the OS will help in loading that executing program into the memory and allocate its resources, and then the program will begin its execution.

Processes

- So, we first have a program which is written and is ready for execution.
- But till that time it is just a passive entity. That means it just sit there without doing anything.
- But the moments it begins execution, at that instance we call that program as a process.
- So, a process can be thought of as a program in execution.
- When a program starts execution, at that time we call it as process.

Processes

- So, when it is not executing, it is just a program sitting there and doing nothing.
- But the moments it begins execution, at that time we call it a process.
- So, in the early computer it supported only one process or one program at a time.
- But in today’s computer, it supports multiple processes or programs running at the same time.
- And even one single program can have many processes associated with it.
- So, when a program begins its execution at that time it is known as the process.

Threads

- So, now let’s try to understand what are threads?
- A thread is the unit of execution within a process.
- A process can have just one thread to many threads.
- So, within a process there may be one or more units of its execution.
- And those units are known as threads.
- In earlier systems, one process had only one thread.
- But now a single process has many threads or many units of execution within it.

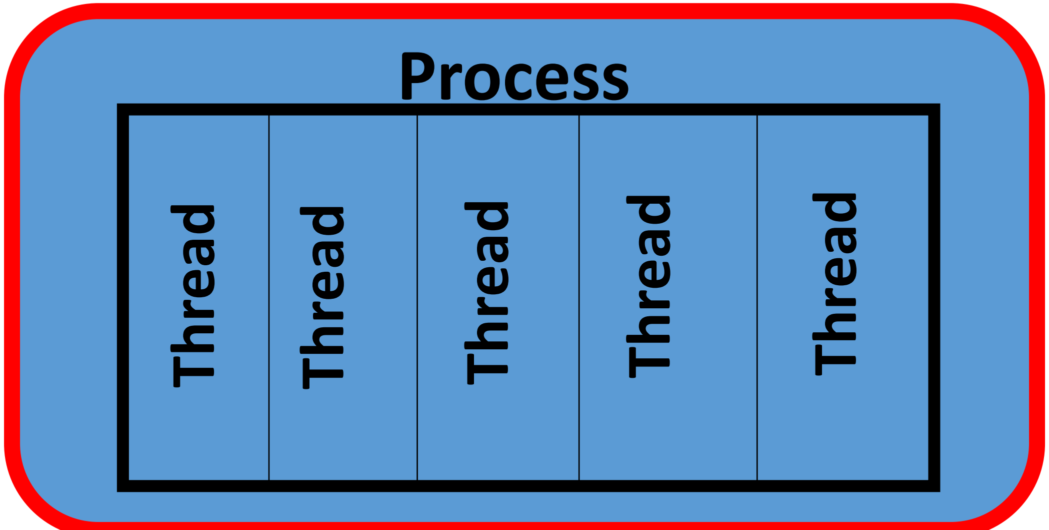
Processes Vs. Threads

- Here we call this entire things as a process.

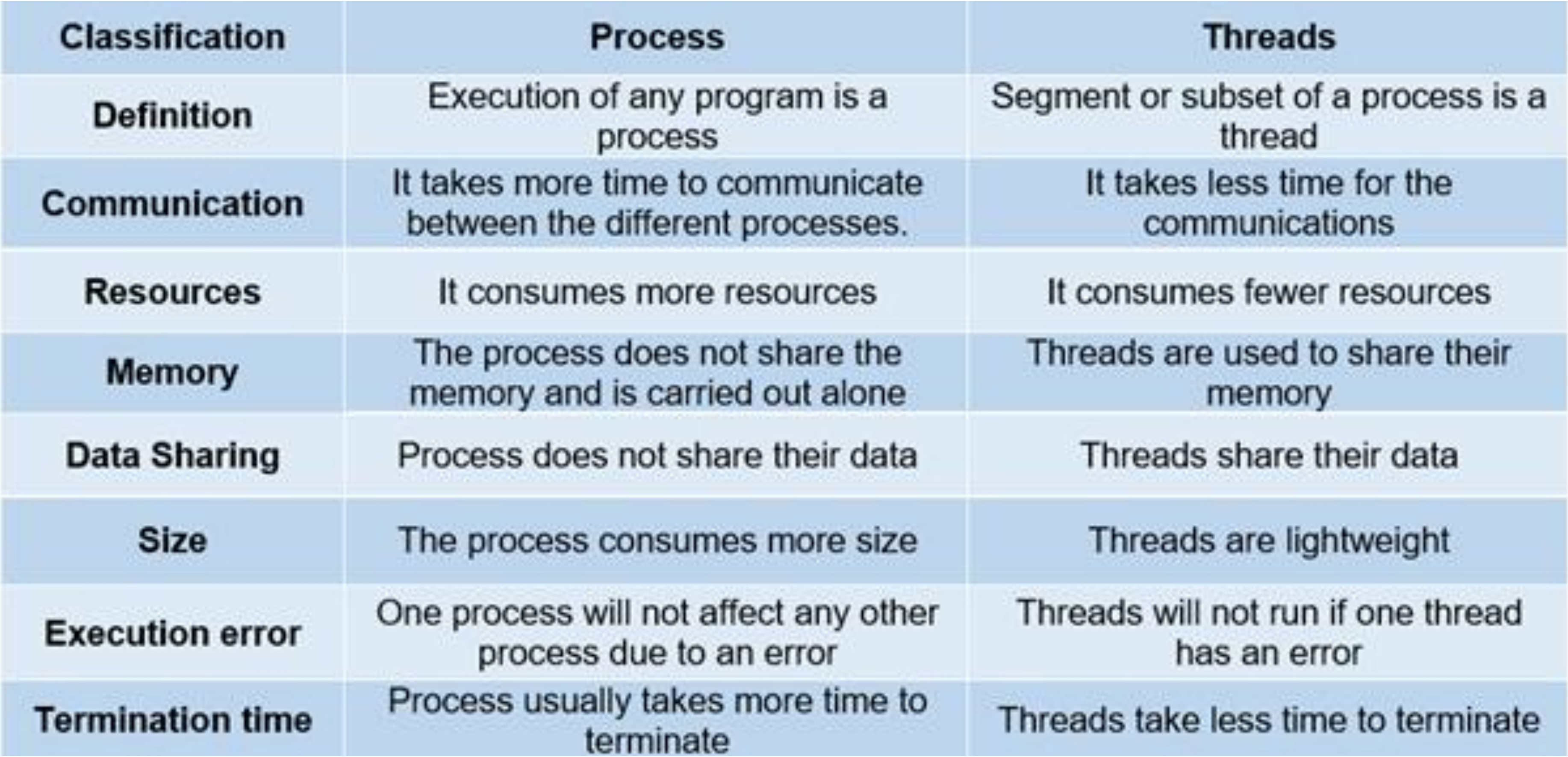
- When a program begins its executions, it becomes the process.

- And within this process there are certain units of executions.

- And each units is known as thread which are under execution within the process.



Processes Vs. Threads

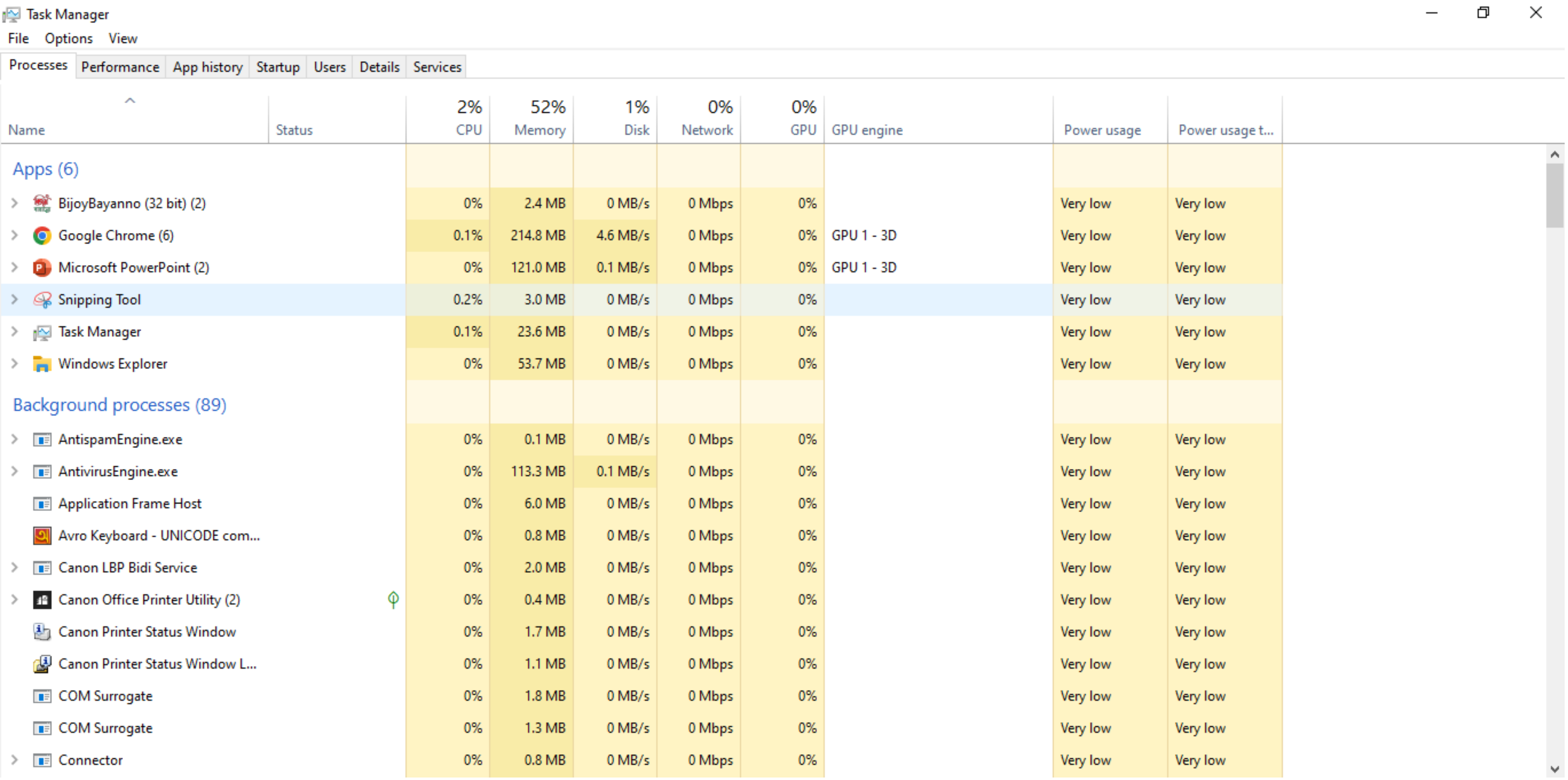


Classification	Process	Threads
Definition	Execution of any program is a process	Segment or subset of a process is a thread
Communication	It takes more time to communicate between the different processes.	It takes less time for the communications
Resources	It consumes more resources	It consumes fewer resources
Memory	The process does not share the memory and is carried out alone	Threads are used to share their memory
Data Sharing	Process does not share their data	Threads share their data
Size	The process consumes more size	Threads are lightweight
Execution error	One process will not affect any other process due to an error	Threads will not run if one thread has an error
Termination time	Process usually takes more time to terminate	Threads take less time to terminate

Processes and Threads

- Now if you want to see the processes which are in execution in your system, you can see it by using your **task manager** in your windows.
- You can open up the task manager and you can see which are the programs that are loaded, and what are the process that are executing.

Processes and Threads



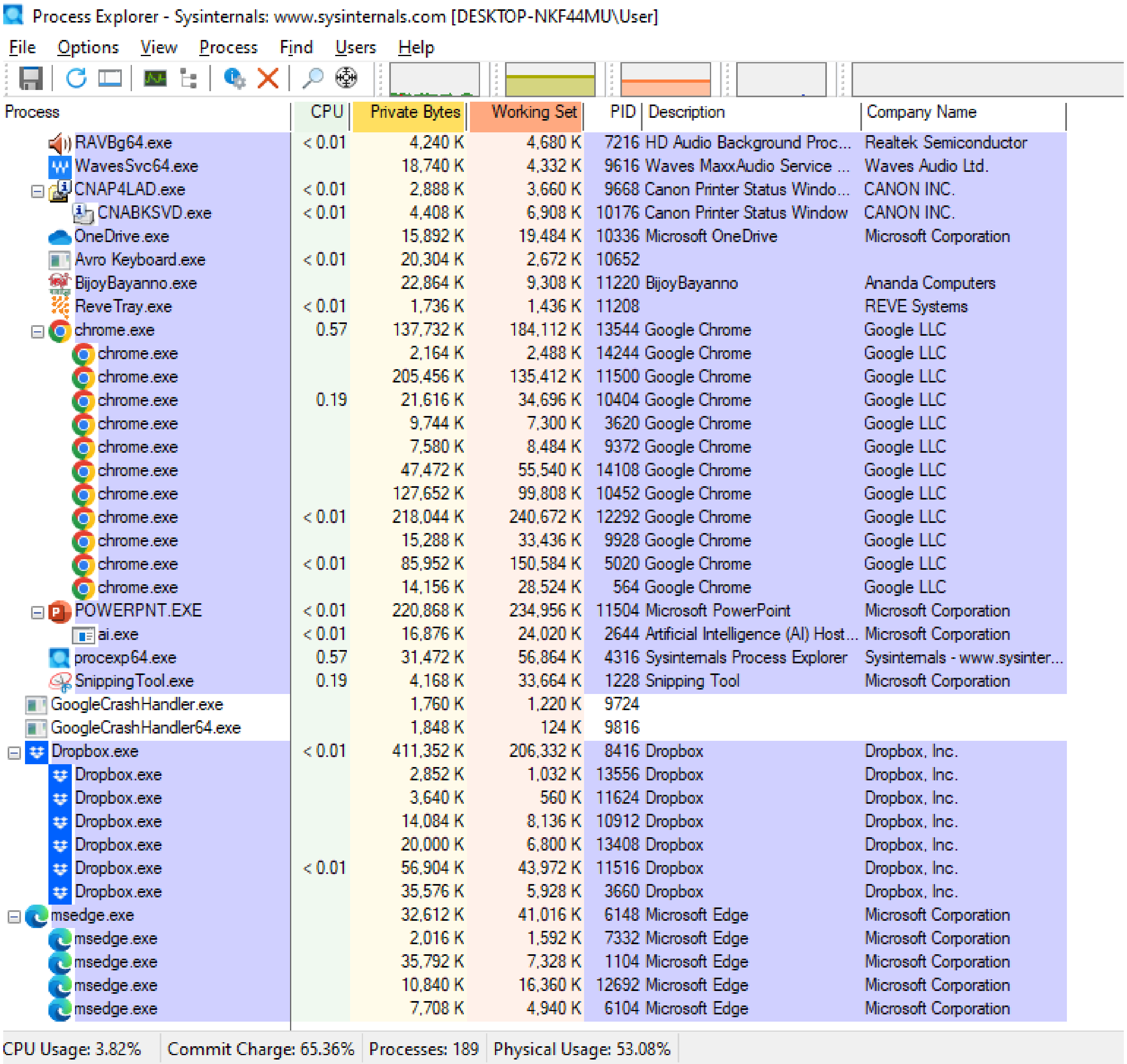
		2%	52%	1%	0%	0%			
Name	Status	CPU	Memory	Disk	Network	GPU	GPU engine	Power usage	Power usage t...
Apps (6)									
BijoyBayanno (32 bit) (2)		0%	2.4 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Google Chrome (6)		0.1%	214.8 MB	4.6 MB/s	0 Mbps	0%	GPU 1 - 3D	Very low	Very low
Microsoft PowerPoint (2)		0%	121.0 MB	0.1 MB/s	0 Mbps	0%	GPU 1 - 3D	Very low	Very low
Snipping Tool		0.2%	3.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Task Manager		0.1%	23.6 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Windows Explorer		0%	53.7 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Background processes (89)									
AntispamEngine.exe		0%	0.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low
AntivirusEngine.exe		0%	113.3 MB	0.1 MB/s	0 Mbps	0%		Very low	Very low
Application Frame Host		0%	6.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Avro Keyboard - UNICODE com...		0%	0.8 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Canon LBP Bidi Service		0%	2.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Canon Office Printer Utility (2)		0%	0.4 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Canon Printer Status Window		0%	1.7 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Canon Printer Status Window L...		0%	1.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low
COM Surrogate		0%	1.8 MB	0 MB/s	0 Mbps	0%		Very low	Very low
COM Surrogate		0%	1.3 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Connector		0%	0.8 MB	0 MB/s	0 Mbps	0%		Very low	Very low

- So, here this is screen shot of my ‘task manager’ (Ctrl+Alt+Del).
- If you open up your task manager and look at Process,
- Here you have seen the processes, but we did not see any threads over here.

Process Explorer

- So, if you want to see the threads that are also there in the system.
- To see the threads, we can use a program known as ***process explorer***.
- You can download that program, and it will show you even the threads that are running for each program and for each process.

- This is how the process explorer looks like.

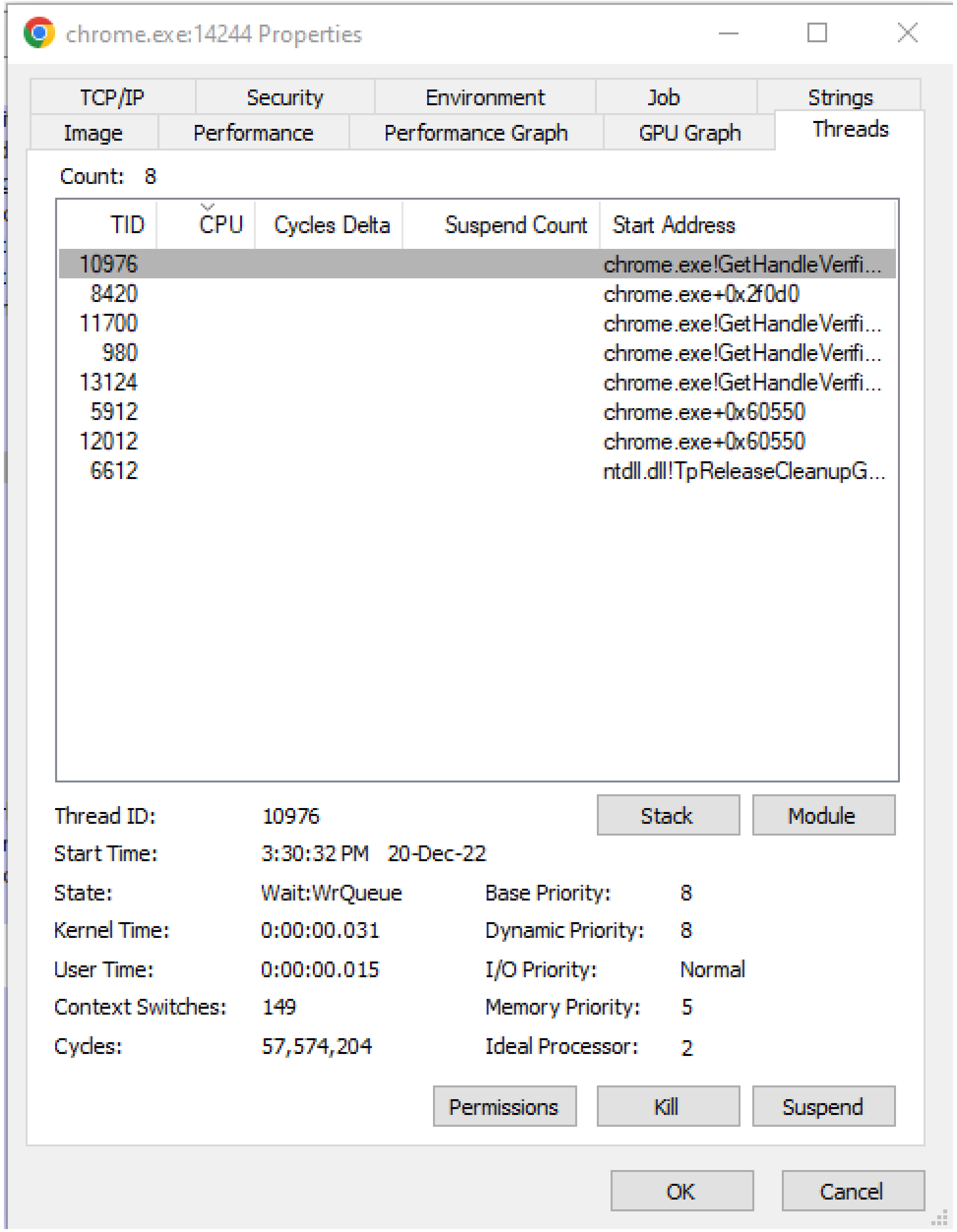


Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-NKF44MU\User]

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
RAVBg64.exe	< 0.01	4,240 K	4,680 K	7216	HD Audio Background Proc...	Realtek Semiconductor
WavesSvc64.exe		18,740 K	4,332 K	9616	Waves MaxxAudio Service ...	Waves Audio Ltd.
CNAP4LAD.exe	< 0.01	2,888 K	3,660 K	9668	Canon Printer Status Windo...	CANON INC.
CNABKSVD.exe	< 0.01	4,408 K	6,908 K	10176	Canon Printer Status Window	CANON INC.
OneDrive.exe		15,892 K	19,484 K	10336	Microsoft OneDrive	Microsoft Corporation
Avro Keyboard.exe	< 0.01	20,304 K	2,672 K	10652		
BijoyBayanno.exe		22,864 K	9,308 K	11220	BijoyBayanno	Ananda Computers
ReveTray.exe	< 0.01	1,736 K	1,436 K	11208		REVE Systems
chrome.exe	0.57	137,732 K	184,112 K	13544	Google Chrome	Google LLC
chrome.exe		2,164 K	2,488 K	14244	Google Chrome	Google LLC
chrome.exe		205,456 K	135,412 K	11500	Google Chrome	Google LLC
chrome.exe	0.19	21,616 K	34,696 K	10404	Google Chrome	Google LLC
chrome.exe		9,744 K	7,300 K	3620	Google Chrome	Google LLC
chrome.exe		7,580 K	8,484 K	9372	Google Chrome	Google LLC
chrome.exe		47,472 K	55,540 K	14108	Google Chrome	Google LLC
chrome.exe		127,652 K	99,808 K	10452	Google Chrome	Google LLC
chrome.exe	< 0.01	218,044 K	240,672 K	12292	Google Chrome	Google LLC
chrome.exe		15,288 K	33,436 K	9928	Google Chrome	Google LLC
chrome.exe	< 0.01	85,952 K	150,584 K	5020	Google Chrome	Google LLC
chrome.exe		14,156 K	28,524 K	564	Google Chrome	Google LLC
POWERPNT.EXE	< 0.01	220,868 K	234,956 K	11504	Microsoft PowerPoint	Microsoft Corporation
ai.exe	< 0.01	16,876 K	24,020 K	2644	Artificial Intelligence (AI) Host...	Microsoft Corporation
procexp64.exe	0.57	31,472 K	56,864 K	4316	Sysinternals Process Explorer	Sysinternals - www.sysinter...
SnippingTool.exe	0.19	4,168 K	33,664 K	1228	Snipping Tool	Microsoft Corporation
GoogleCrashHandler.exe		1,760 K	1,220 K	9724		
GoogleCrashHandler64.exe		1,848 K	124 K	9816		
Dropbox.exe	< 0.01	411,352 K	206,332 K	8416	Dropbox	Dropbox, Inc.
Dropbox.exe		2,852 K	1,032 K	13556	Dropbox	Dropbox, Inc.
Dropbox.exe		3,640 K	560 K	11624	Dropbox	Dropbox, Inc.
Dropbox.exe		14,084 K	8,136 K	10912	Dropbox	Dropbox, Inc.
Dropbox.exe		20,000 K	6,800 K	13408	Dropbox	Dropbox, Inc.
Dropbox.exe	< 0.01	56,904 K	43,972 K	11516	Dropbox	Dropbox, Inc.
Dropbox.exe		35,576 K	5,928 K	3660	Dropbox	Dropbox, Inc.
msedge.exe		32,612 K	41,016 K	6148	Microsoft Edge	Microsoft Corporation
msedge.exe		2,016 K	1,592 K	7332	Microsoft Edge	Microsoft Corporation
msedge.exe		35,792 K	7,328 K	1104	Microsoft Edge	Microsoft Corporation
msedge.exe		10,840 K	16,360 K	12692	Microsoft Edge	Microsoft Corporation
msedge.exe		7,708 K	4,940 K	6104	Microsoft Edge	Microsoft Corporation

CPU Usage: 3.82% | Commit Charge: 65.36% | Processes: 189 | Physical Usage: 53.08%

- **This is how the Threads for Google chrome looks like.**



chrome.exe:14244 Properties

Count: 8

TID	CPU	Cycles Delta	Suspend Count	Start Address
10976				chrome.exe!GetHandleVerifi...
8420				chrome.exe+0x2f0d0
11700				chrome.exe!GetHandleVerifi...
980				chrome.exe!GetHandleVerifi...
13124				chrome.exe!GetHandleVerifi...
5912				chrome.exe+0x60550
12012				chrome.exe+0x60550
6612				ntdll.dll!TpReleaseCleanupG...

Thread ID:	10976		
Start Time:	3:30:32 PM 20-Dec-22		
State:	Wait:WrQueue	Base Priority:	8
Kernel Time:	0:00:00.031	Dynamic Priority:	8
User Time:	0:00:00.015	I/O Priority:	Normal
Context Switches:	149	Memory Priority:	5
Cycles:	57,574,204	Ideal Processor:	2

Relation between Program and Process in OS

- It is same with different name ,
- when this is sleeping (not executing) it is called program and
- when it is executing becomes process.
- Well, to be very precise. Process is not the same as program.
- A process is more than a program code.
- A process is an ‘active’ entity as oppose to program which is considered to be a ‘passive’ entity.
- As you all know that a program is an algorithm expressed with the help of a programming language.

Relation between Program and Process in OS

- A program is a passive entity sitting on some secondary storage device.
- Process, on the other hand, includes:
  - Current value of Program Counter (PC)
  - Contents of the processors registers
  - Value of the variables
  - The process-stack (SP) which typically contains temporary data such as subroutine parameter, return address, and temporary variables.
  - A data section that contains global variables.
  - A process is the unit of work in a system .

Process State

- Previously, we knew that a program in execution is know as process.
- As a process executes, it changes state.
- The state of a process is defined in part by the current activities of that process.
- What is the process currently doing?
- So, when a process is executing, it can change from one state to another.
- Now, let us see what are the states that a process can be in.

Process State

- Each process may be in one of the following states:

New	**The process is being created**
Running	**The 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**
Ready	**The process is waiting to be assigned to a processor**
Terminated	**The process has finished execution.**

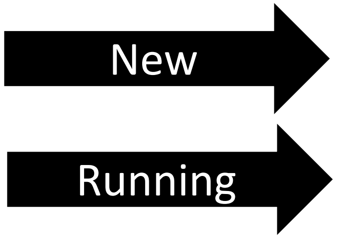
Process State

New

The process is being created

- So, first of all, we have “New”.
- So, if a process is in “New” state, it means the process is being created.
- When the process is being created for performing a particular task. at that state or at that moment, we say that the process is in ‘new’ state.

Process State



New

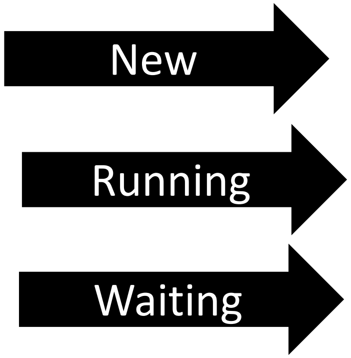
The process is being created

Running

The instructions are being executed

- •Now, the second state is ‘Running state’.
- •So, instruction are being executed at that moment, we say that the process is in the ‘Running ’ state.
- •So, when the process is being created, it is know as ‘New’ state.
- •And when the process begins its execution, that means when the instruction within the process are being executed. At that moment we say that the process is in the ‘Running ’ state.

Process State



New

The process is being created

Running

The 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

•There is another state know as the ‘Waiting’ state.

•So, the process is waiting for some event to occur.

•Such as an I/O completion or reception of a signal.

•So, we say that a process is in the ‘Waiting’ state, when it is waiting for some event to occur.

•Some I/O operation needs to be completed which are in high priority.

Process State

New	**The process is being created**
Running	**The 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**
Ready	**The process is waiting to be assigned to a processor**

- The process is waiting to be assigned to a processor . So, when the process has been created that means this process is already being created, the ‘New’ state is already over, and it is not yet running.
- So, after the process is created and before it starts running, the process has to be assigned to a processor so that it can start running or it can start its execution.
- So, at that moment we say that the process is in ‘Ready’ state.

Process State

New	**The process is being created**
Running	**The 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**
Ready	**The process is waiting to be assigned to a processor**
Terminated	**The process has finished execution.**

- The then the last one is called the ‘Terminated’ state.
- So, from the name itself you must have understood the process has finished its execution.
- So, when the process finishes its execution, then we say that it is in the ‘Terminated’ state or the process has been terminated.

Process State

- New
- Running
- Waiting
- Ready
- Terminated

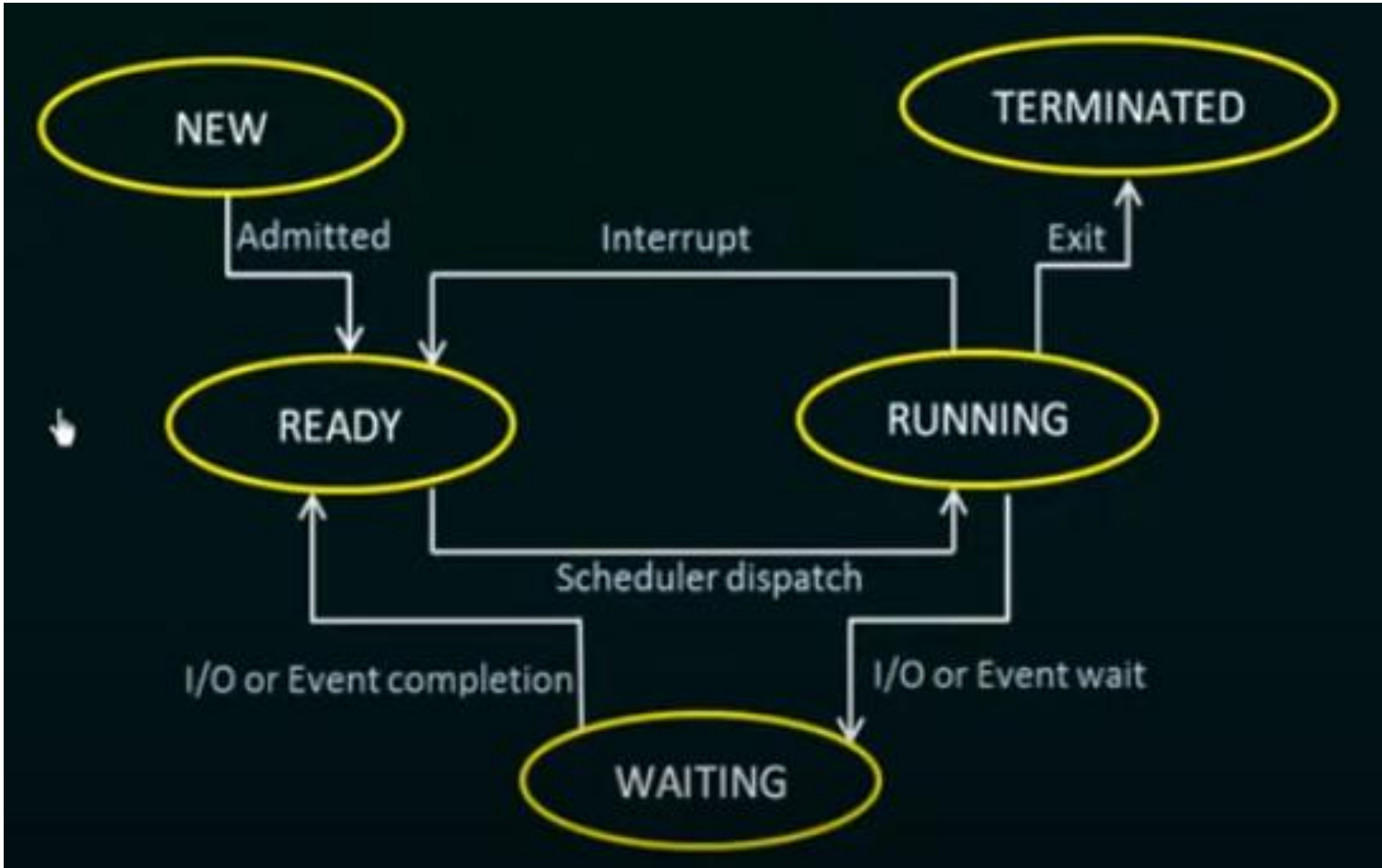


Diagram of Process State

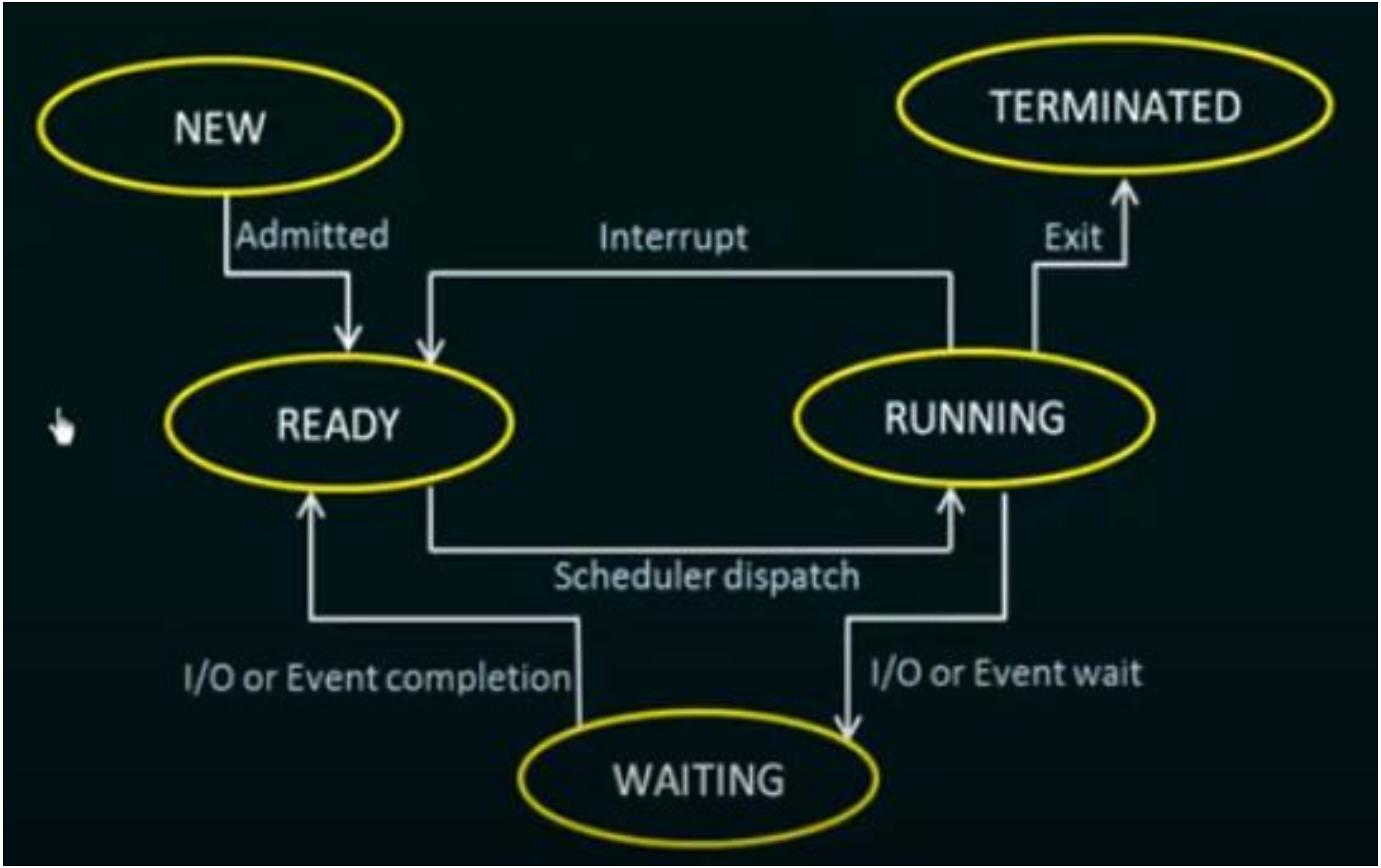
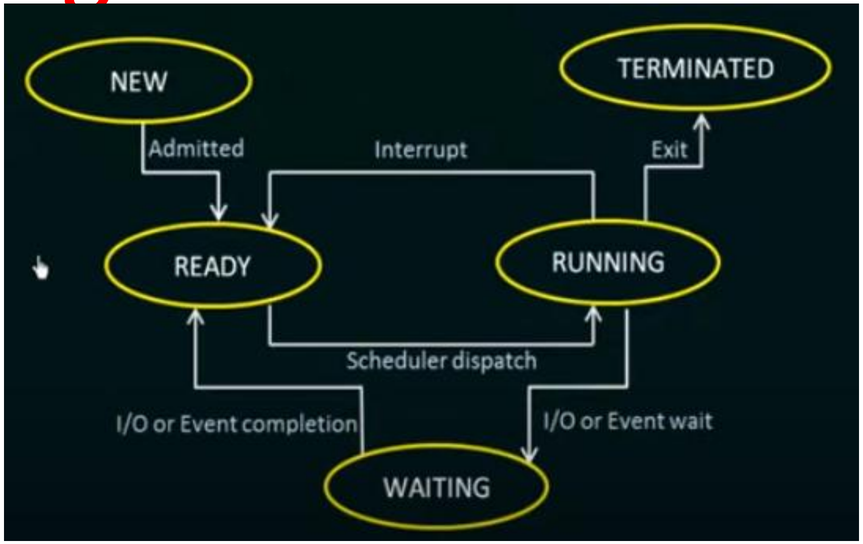


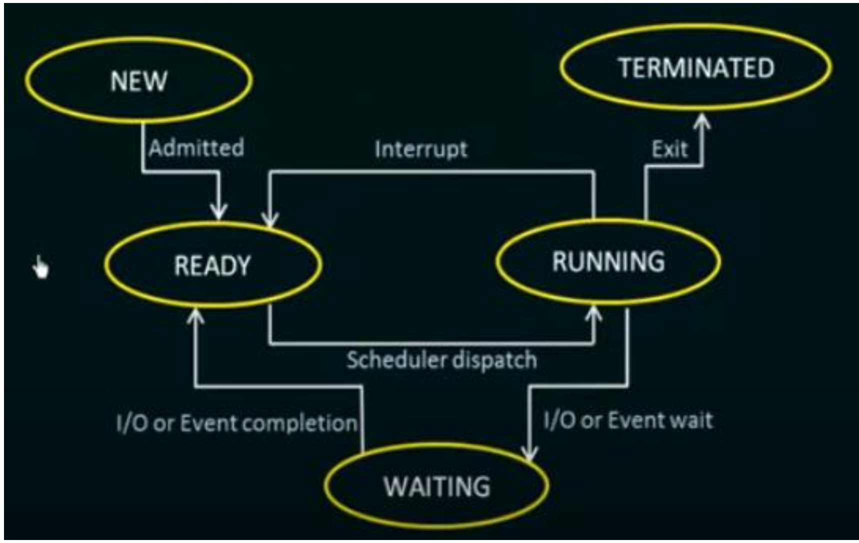
Diagram of Process State



- Let us see how does a process change from one state to another using this diagram.
- So, first of all, we have the ‘New’ state. here the process is being created.
- So, when the process is being created , it is there in this ‘New’ state.
- After the process is created, it is admitted and it comes to the ‘Ready’ state.

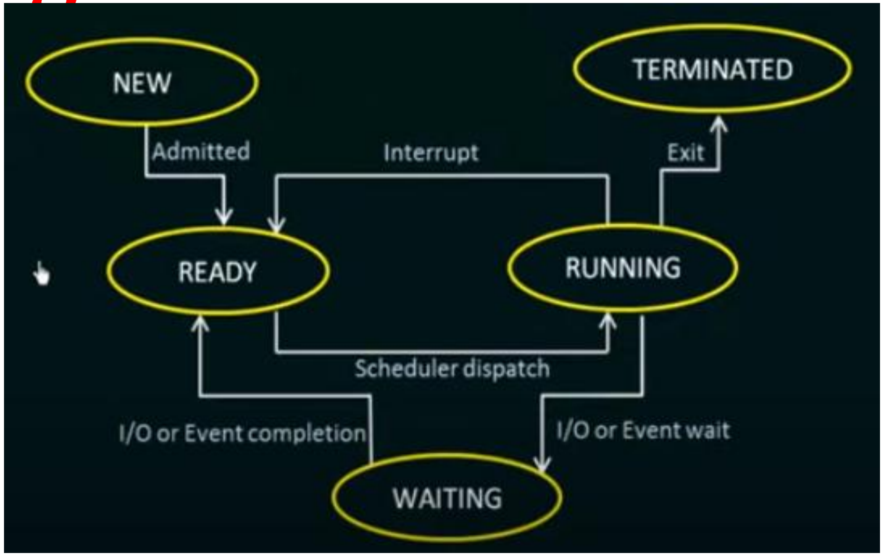
Diagram of Process State

- In the ‘Ready’ state, the process is ready to begin execution, and it is waiting to be assigned to a processor so that it can begin its execution.



- So, a process is created, admitted, and ready over here.
- In ready state, what can happen?
- The scheduler dispatches it to the processor, and then when it begins its execution, it is in the ‘Running’ state.
- When it is executing or when the instruction in the process is being executed, at that moment we say that it is in the ‘Running’ state.

Diagram of Process State

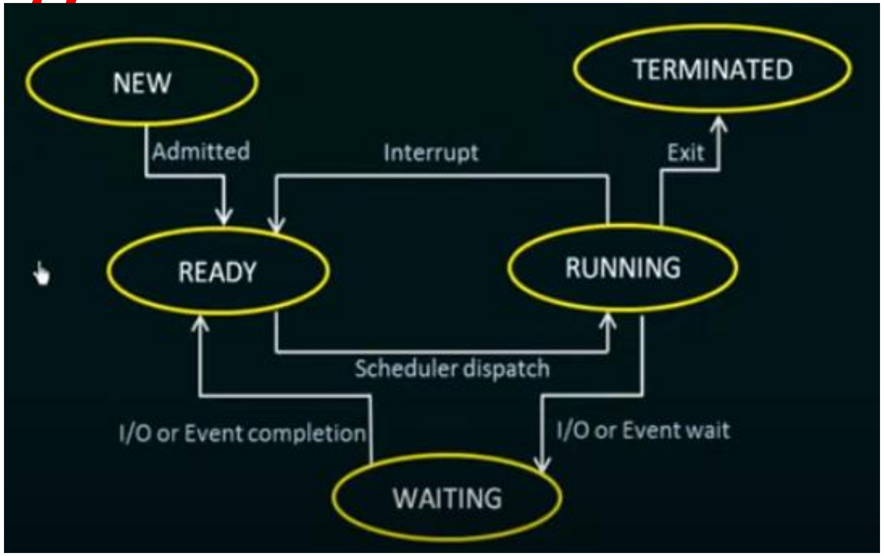


- Now after coming to the ‘Running’ state, there are **THREE** cases that can occur.

- So, **the first case**, it was running, and it finished its executions. And then it exit and goes to the ‘Terminated’ state.

- That means there was no interruptions in its running and it was properly executed. And once it finishes its execution, it exits and goes to the terminated stage.

Diagram of Process State



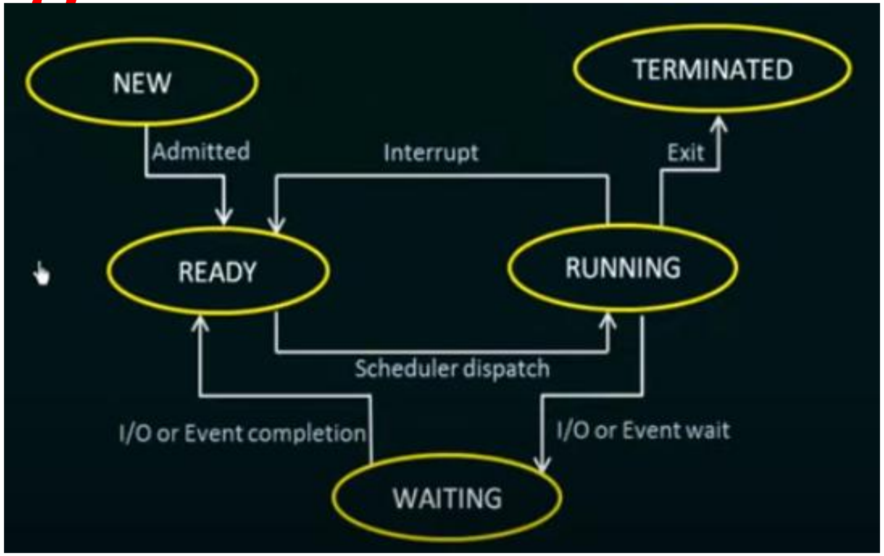
•**In the next case** that can occur is there can be a interrupt when the process was being executed.

•When the process was being executed, some other process with high priority came, and this process was interrupted.

•So, when it gets interrupted, it goes back to the ‘Ready ‘ state again.

•And once an interrupt has been handled , this process an again be dispatched.

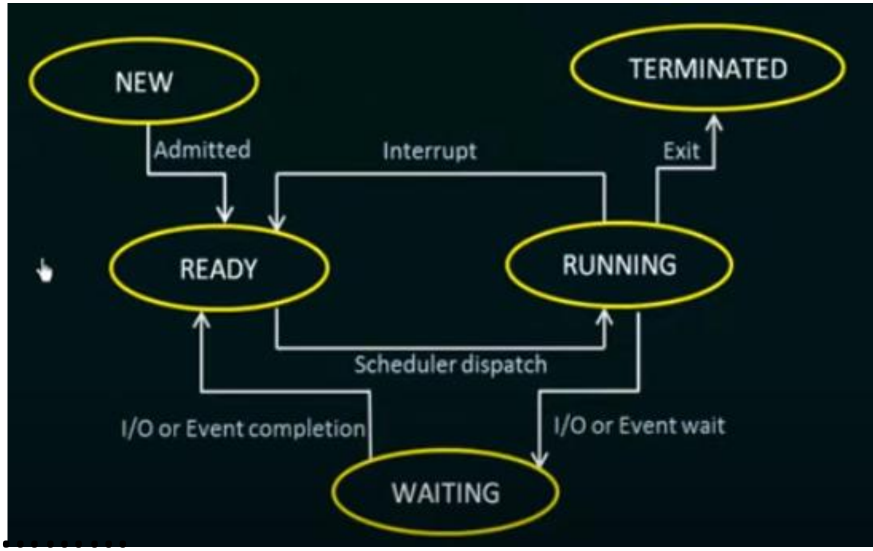
Diagram of Process State



•And assigned to a processor and it can begin its execution. And go to the ‘Running’ state again.

•So, that is what happen if an interrupt is encountered .

Diagram of Process State



- **Now the next case** that can occur…………

- When it is running, it needs to wait for an I/O operation or an event.

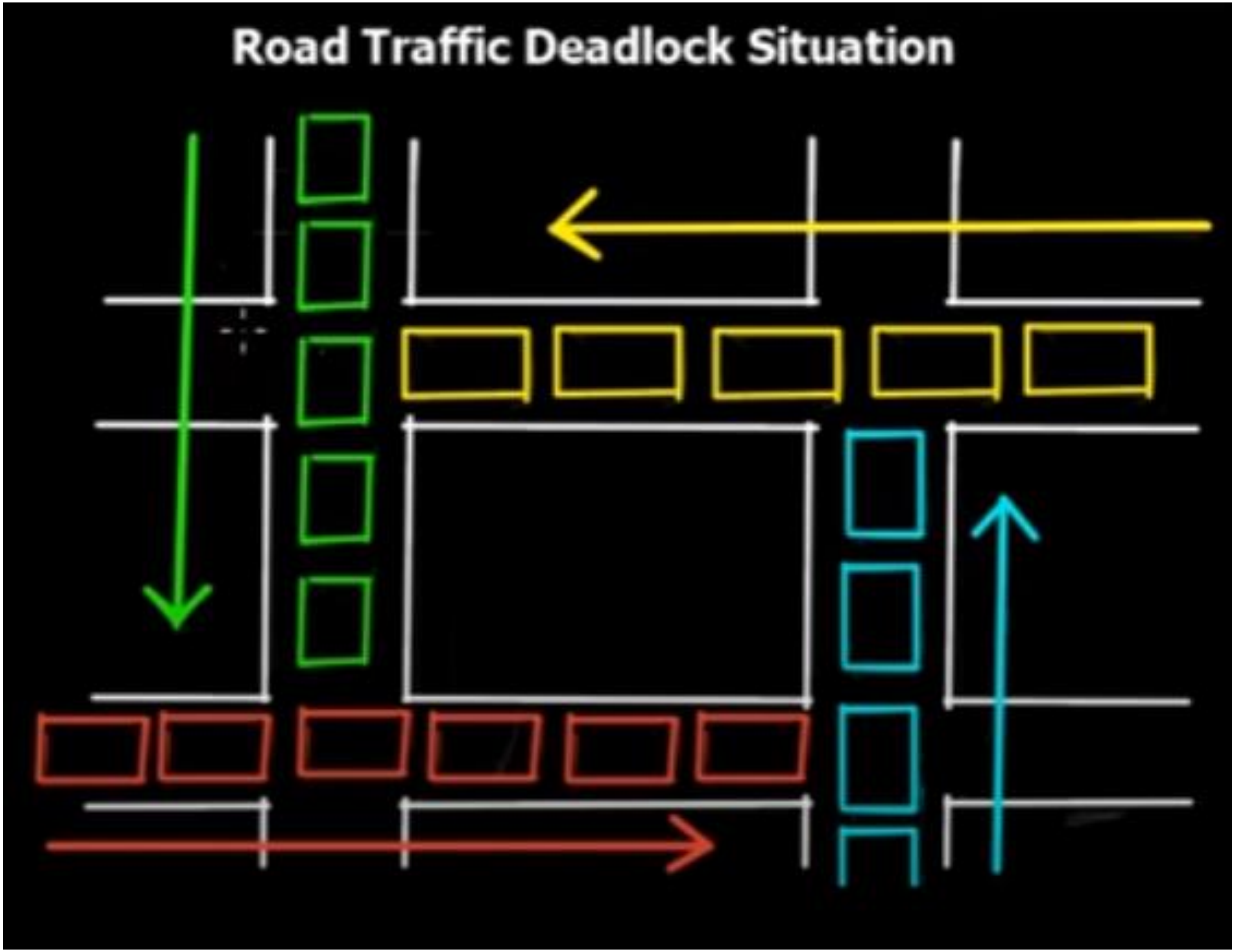
- So, lets say that this process needs some I/O devices.

- At that time, what happens?

- It goes to the ‘Waiting ’ state where it is waiting for an I/O or an event to occur.

- Once, an I/O or an event that it was waiting for has been completed. It goes to the ‘Ready’ state again. So, it is ready to be executed again.

Deadlock



Deadlock?

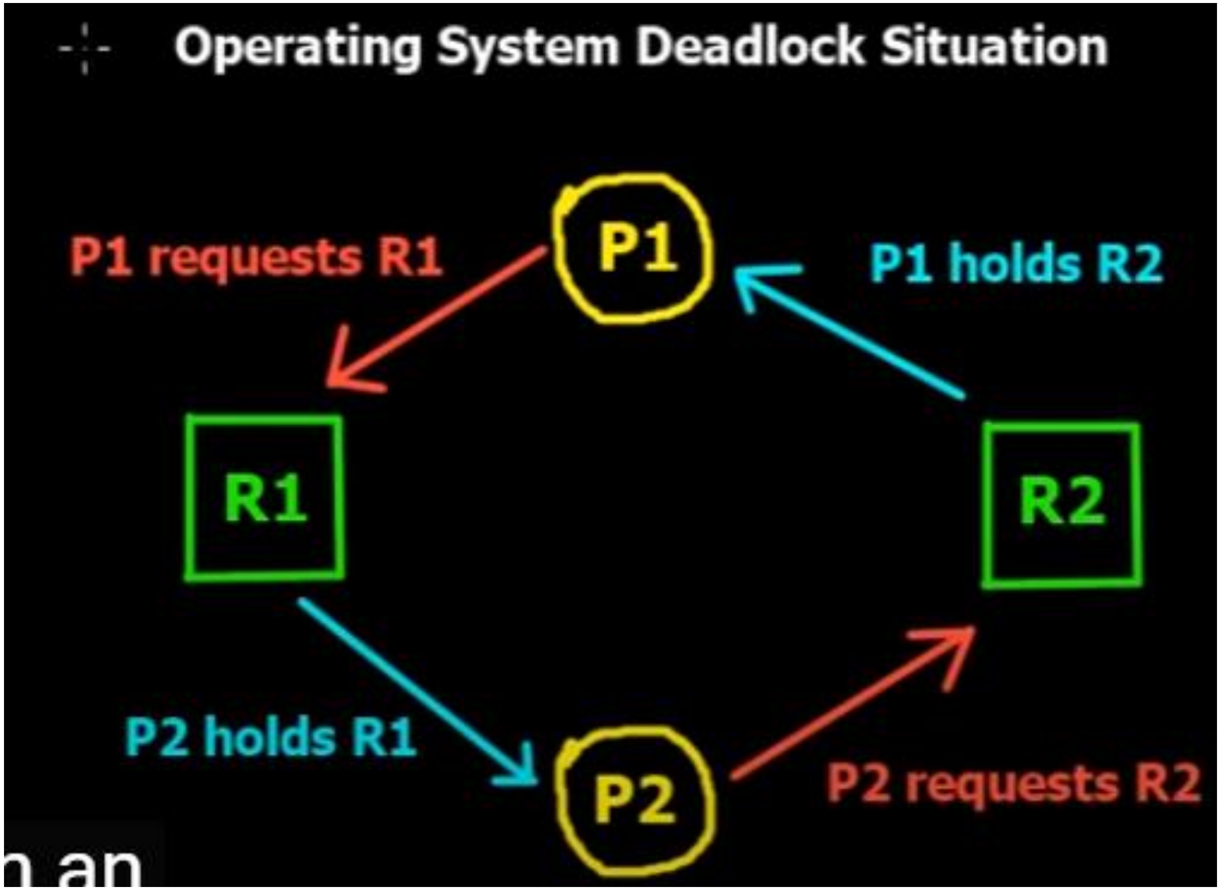
- What is deadlock?
- How does a deadlock happen?
- Real world vs OS scenario of deadlock?
- 4 conditions for a deadlock to occur in OS?
- Deadlock handling strategies?

Before knowing about Deadlock in OS

- A Process in OS uses different resources and uses resources in the following ways:
    - Request a resources
    - Use the resources
    - Releases the resource
- Resource in a computer system can be……
    - Files
    - Databases
    - Other processes
    - I/O
    - Library files
    - Hardware access, etc.

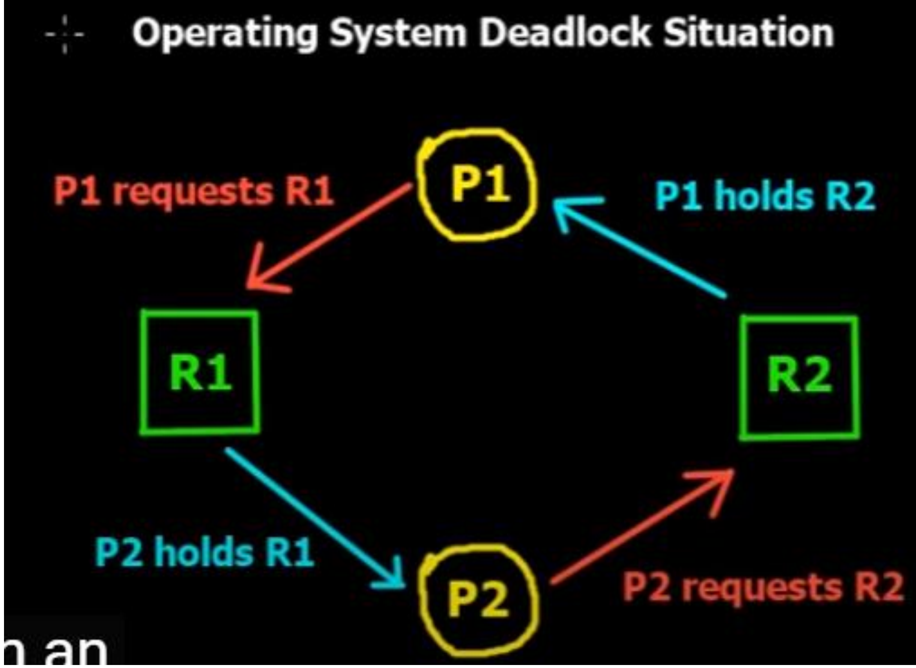
What is Deadlock in OS ?

- Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.



Deadlock in OS

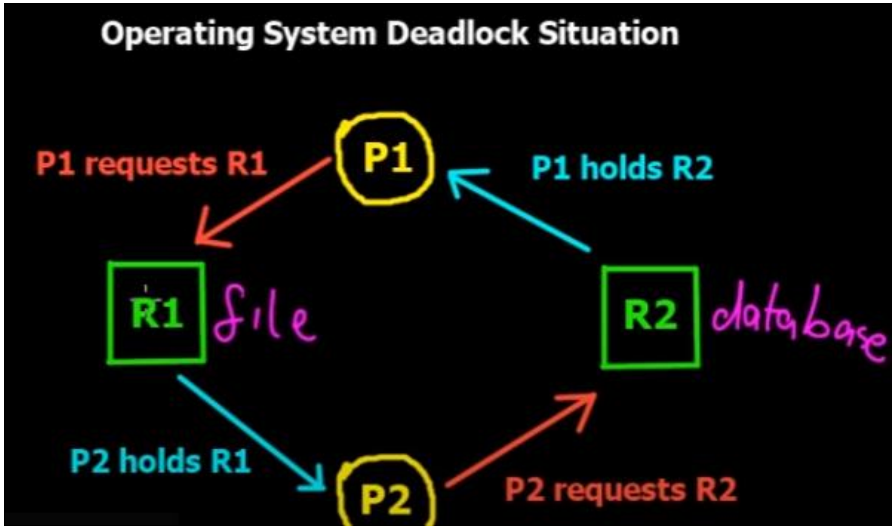
- It’s a very simple diagram to illustrate deadlock in OS.
- In the yellow circle we process P1 and Process P2.



- So, this P1 is process-1 and P2 is process 2.
- In the green square boxes we have resource- R1 and R2.
- This resources might be something like files, data bases, and so on.

Deadlock in OS

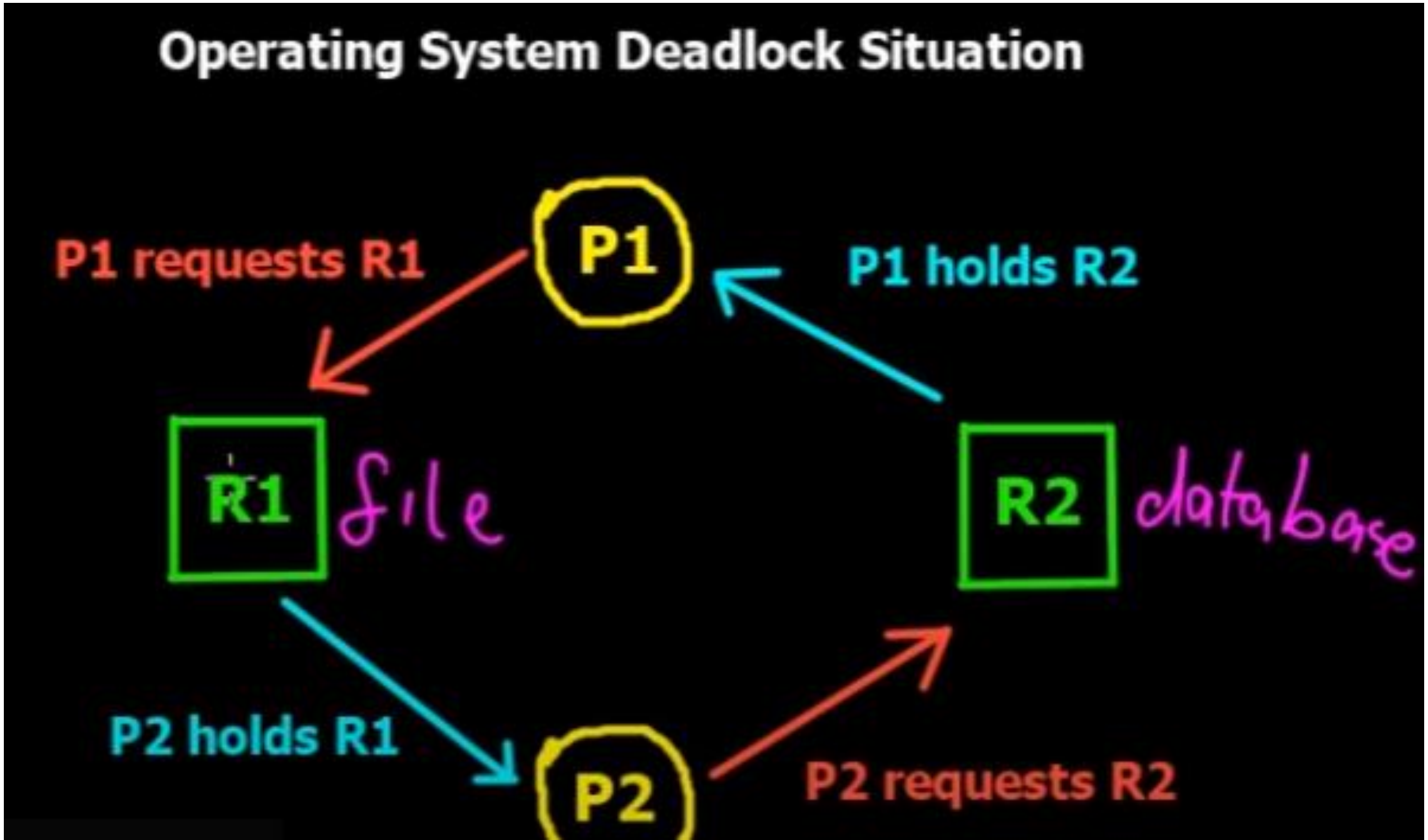
- Now, here’s a situation P1 holds R2 which means that this database R2 is already assigned to P1.



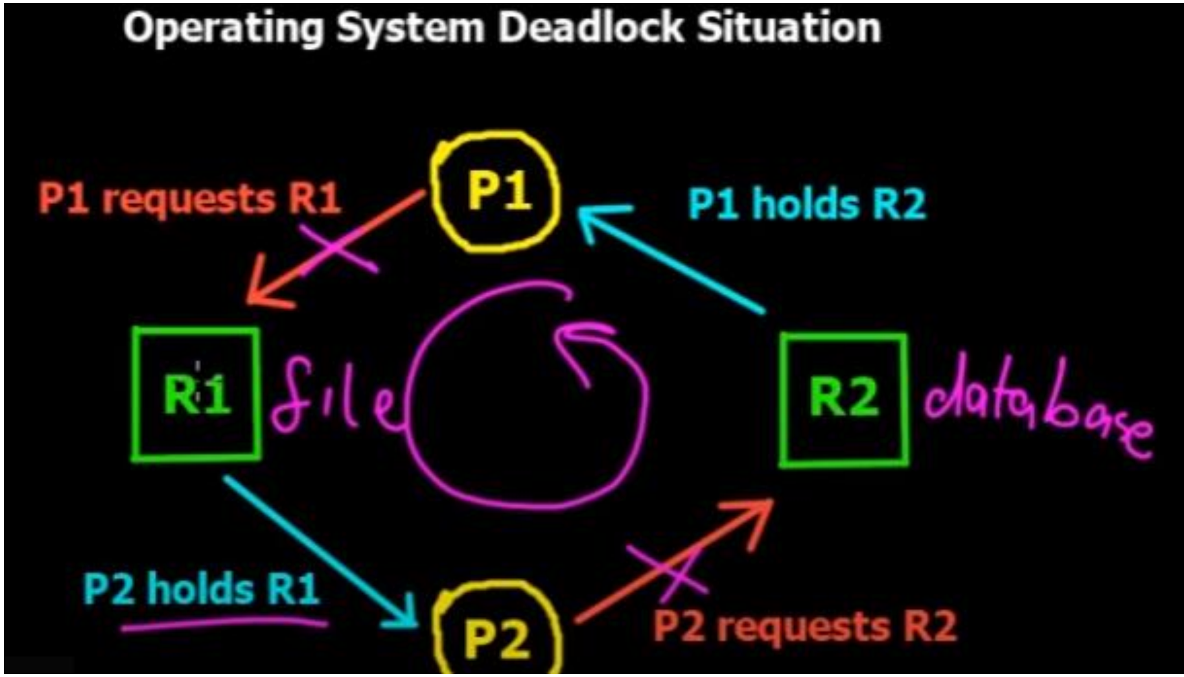
- So, P1 has complete access to R2 which is database. But P1 now wants R1 which is a file.
- So, lets say P1 is a process which copies certain values from the databases R2 and is going to print it in the files, R1 so that’s why P1 wants access of the files but P1 is not able to access R1 because R1 that is this file is allocated to P2.
- P1 holds R2

Deadlock in OS

- Now, P2 wants control of the database that is the R2.
- So, P2 is request for R2 but P2 can not access R2 because R2 is assigned to P1 .

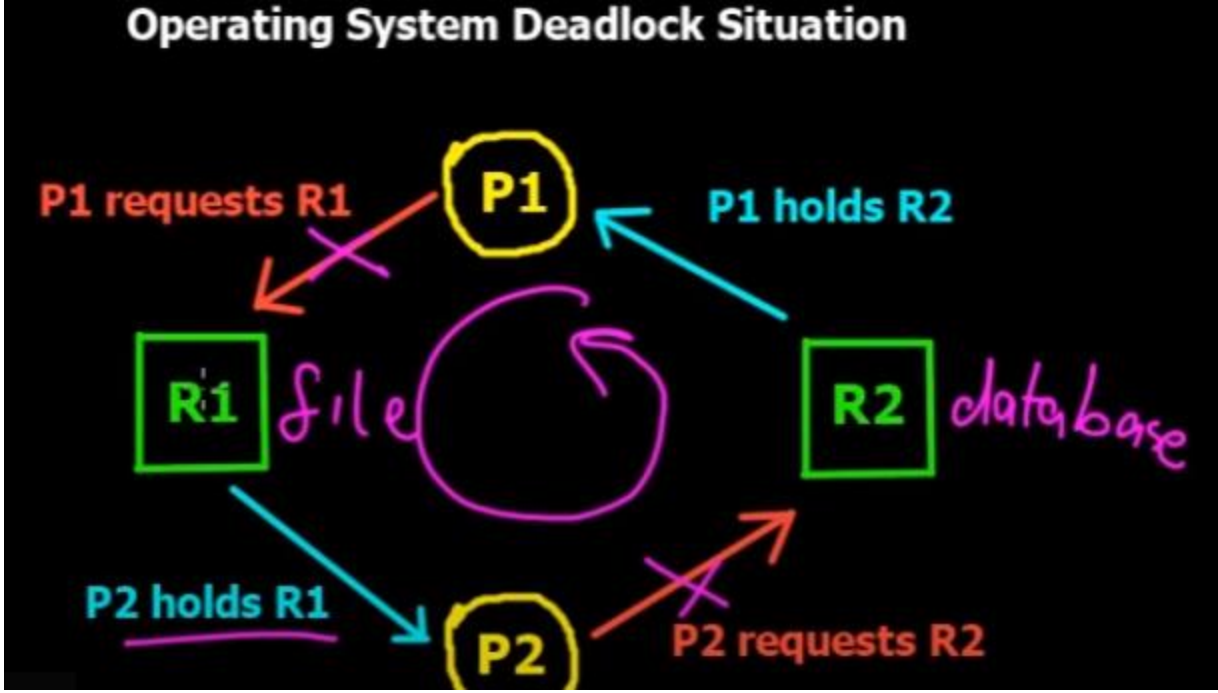


Deadlock in OS



- So, we can see that there is another circular pattern going on over here…
- P1 can not access R1 and P2 can not access R2 because each of them are assigned to either of the two processes respectively that is P1 has control to R2 and P2 has access of R1.

Deadlock in OS



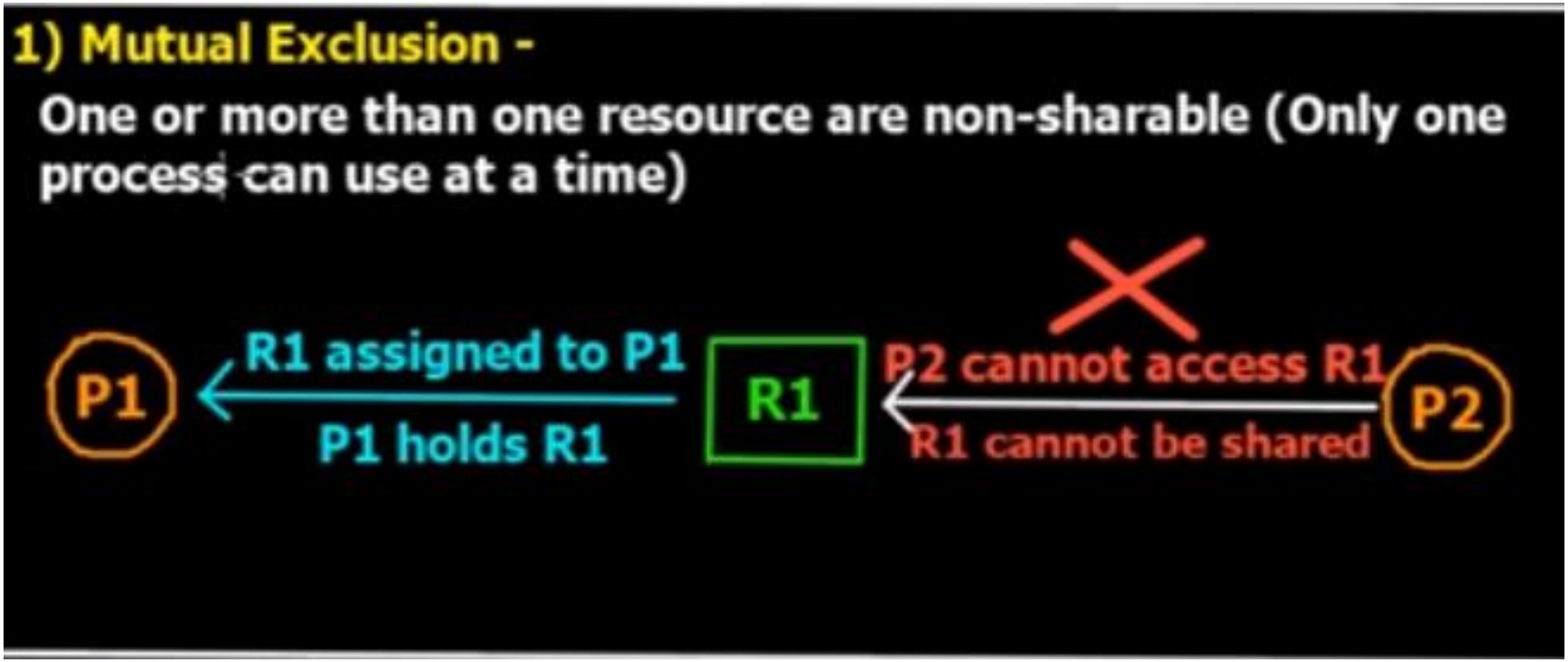
- So both of them are in deadlock situation wherein they are waiting for each other to release those resources so that they can access it so this was a deadlock situation a very basic deadlock situation in our OS.

Conditions for Deadlock to Occur

- Deadlock can arise if following FOUR conditions hold simultaneously (Necessary Conditions) –
  - Mutual Exclusion
  - Hold and Wait
  - No Preemption
  - Circular wait.
- These are very important conditions which is definitely need to understand each of which will be discussed now.

Conditions for Deadlock to Occur

- Deadlock can arise if following FOUR conditions hold simultaneously (Necessary Conditions) –**Mutual Exclusion**



1) Mutual Exclusion -

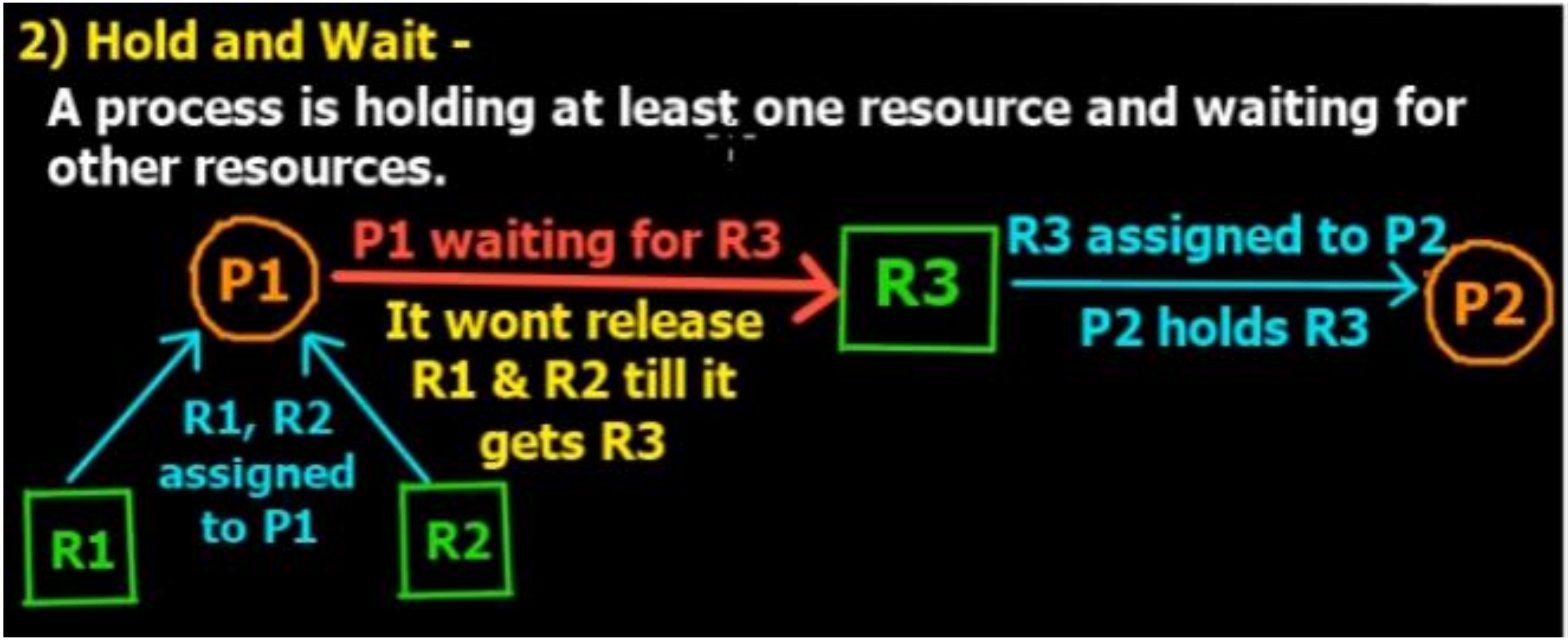
One or more than one resource are non-sharable (Only one process can use at a time)

•Until and unless P1 releases the access of R1, P2 can not access it right P2 is going to be waiting for R1.

•This can lead to a deadlock situation.

Conditions for Deadlock to Occur

- Deadlock can arise if following FOUR conditions hold simultaneously (Necessary Conditions) –**Hold and Wait**



2) Hold and Wait -

A process is holding at least one resource and waiting for other resources.

P1 waiting for R3

It wont release R1 & R2 till it gets R3

R3 assigned to P2

P2 holds R3

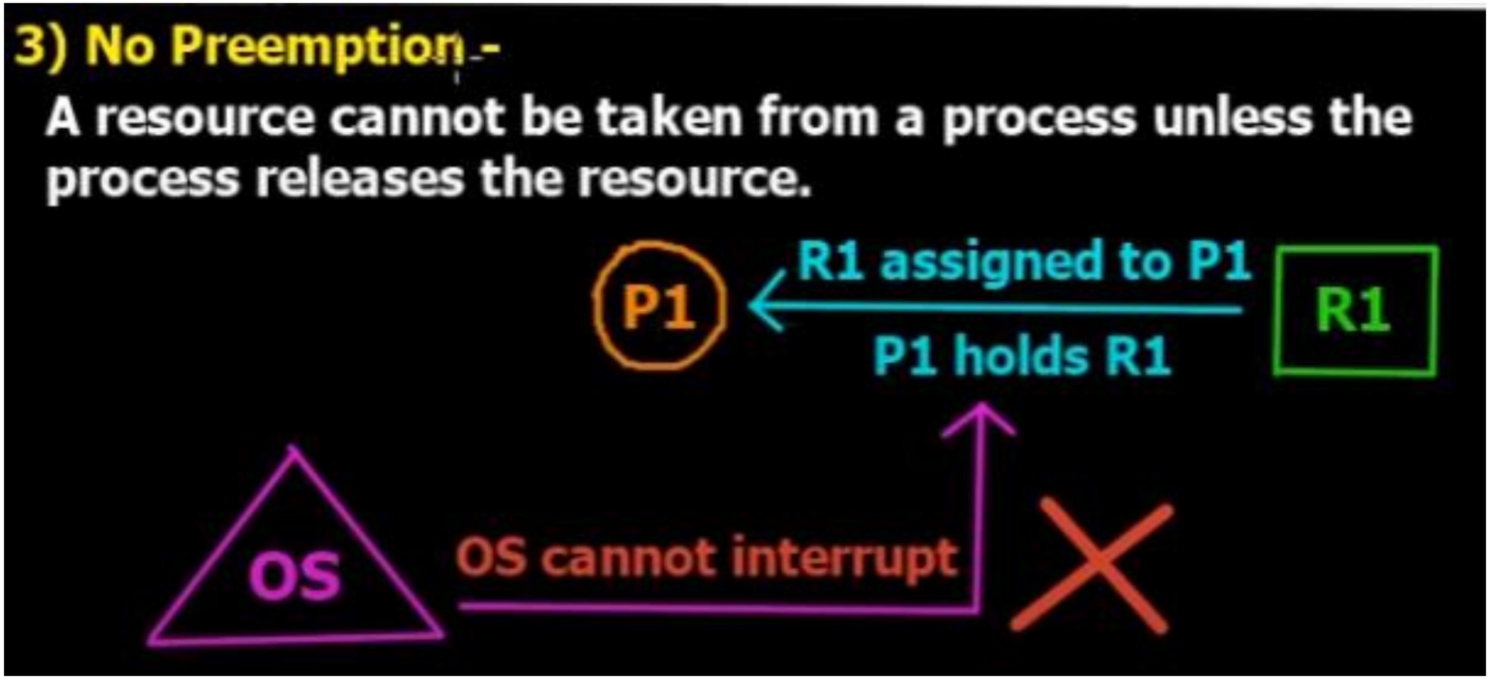
R1, R2 assigned to P1

•P1 goes into deadlock situation of waiting for R3 and P2also goes in a situation for waiting for R1 and R2 because of one resource being held by P1.

- This lead to hold and wait deadlock

Conditions for Deadlock to Occur

- Deadlock can arise if following FOUR conditions hold simultaneously (Necessary Conditions) –**No Preemption**



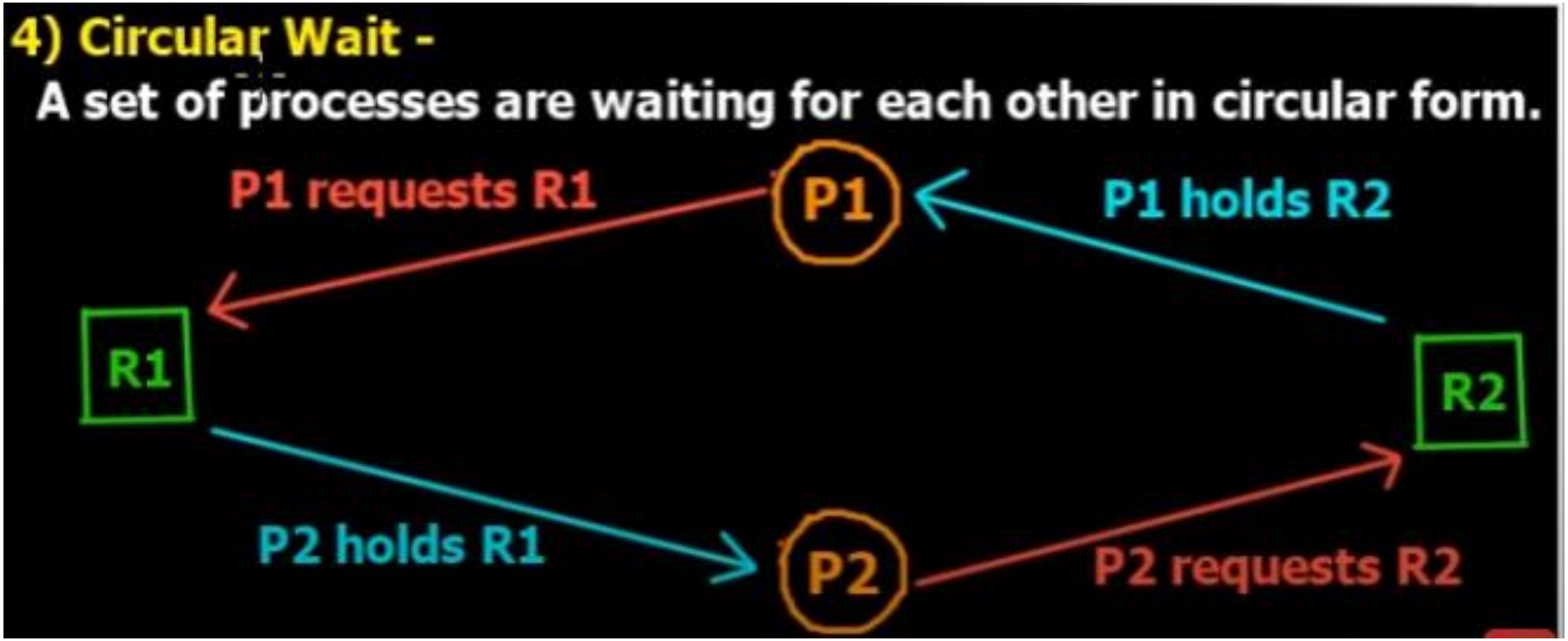
3) No Preemption -

A resource cannot be taken from a process unless the process releases the resource.

- Preemption means pre-empting a particular process which means stopping a process in between.
- No action can be performed even by OS to stop any activities in between, so this states that a resource can not be taken from a process unless the process releases the resource.

Conditions for Deadlock to Occur

- Deadlock can arise if following FOUR conditions hold simultaneously (Necessary Conditions) –**Circular Wait**



4) Circular Wait -

A set of processes are waiting for each other in circular form.

Methods for Handling deadlock

There are three ways to handle deadlock:

- Deadlock prevention or avoidance-
  - The idea is to not let the system into deadlock state.
- Deadlock detection and recovery-
  - Let deadlock occur, then do preemption to handle it once occurred.
- Ignore the problem all together-
  - If deadlock is very rare, then let it happen and reboot the system
  - Ignore the problem and pretend that deadlocks never occur in the system

Strategies to Handling deadlocks

There are three strategies to handle deadlock:

- Preemption-
  - —We can take a resource from one process and give it to other.
  - —This will resolve the deadlock situation, but sometimes it does causes problems.
- Rollback-
  - —In situations where deadlock is a real possibility, the system can periodically make a record of the state of each process and when deadlock occurs, roll everything back to the last checkpoint, and restart, but allocating resources differently so that deadlock does not occur.
  - —It lead to loss of progresses.
- Kill one or more processes-
  - —Whichever processes are in deadlock you probably just kill them,. This is the simplest way.
  - —You have to restart the process and you can not guarantee that again deadlock will not happened.

Methods Vs. Strategies

- A method is a process, a procedure, or a way something is done or implemented.
- On the other hand, a strategy is a goal, a set of actions, or plans to achieve one aim.

- For example,
  - — if you were planning to study for your exams, then your strategy would be to read all books related to the topic,
  - —while your method would be to follow the timetable provided by the university.