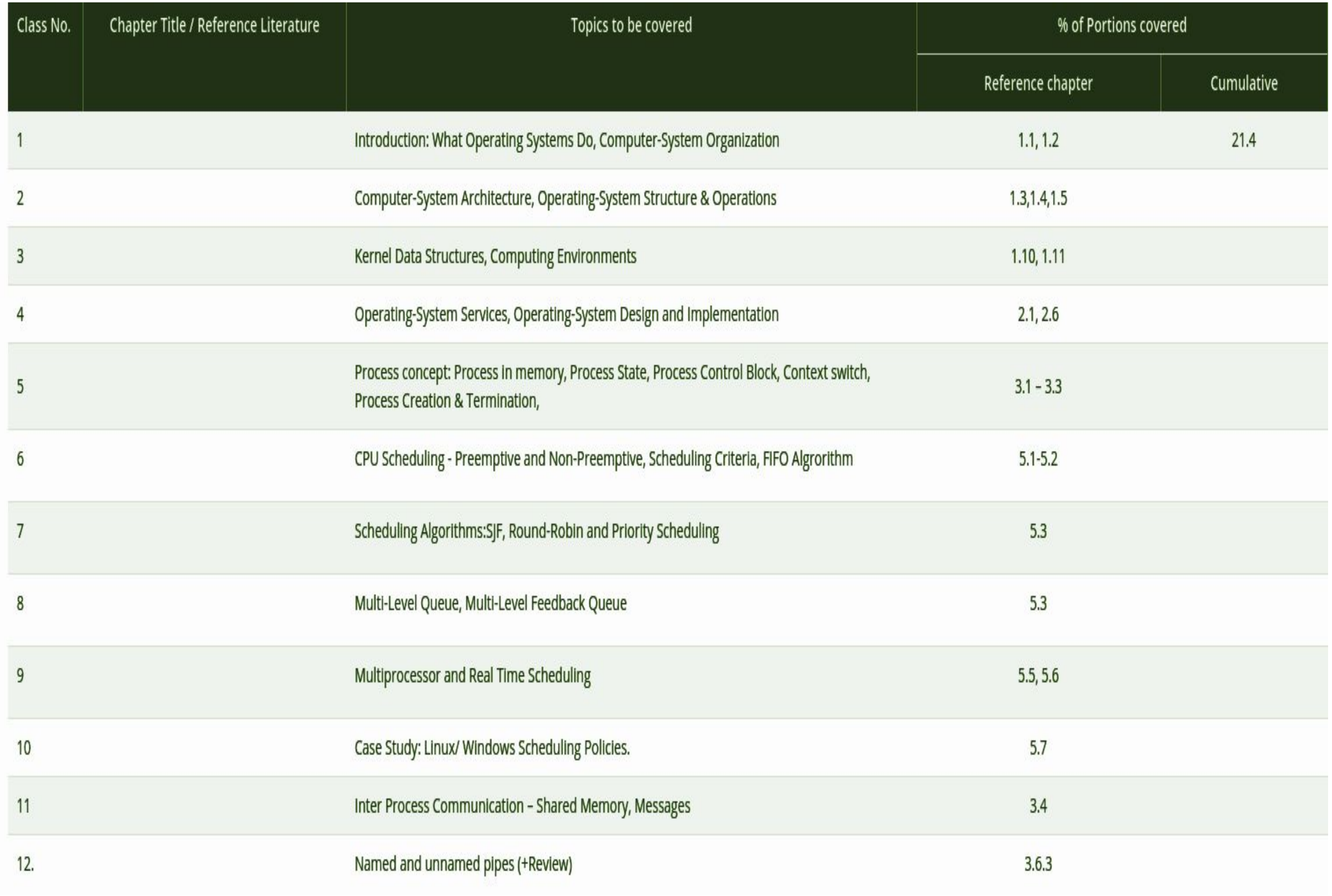
OPERATING SYSTEMS

Unit1_Unit2_Unit3:Revision Class #8 Typical Concepts and QnAs related to Operating Systems

Nitin V Pujari
Faculty, Computer Science
Dean , IQAC, PES University

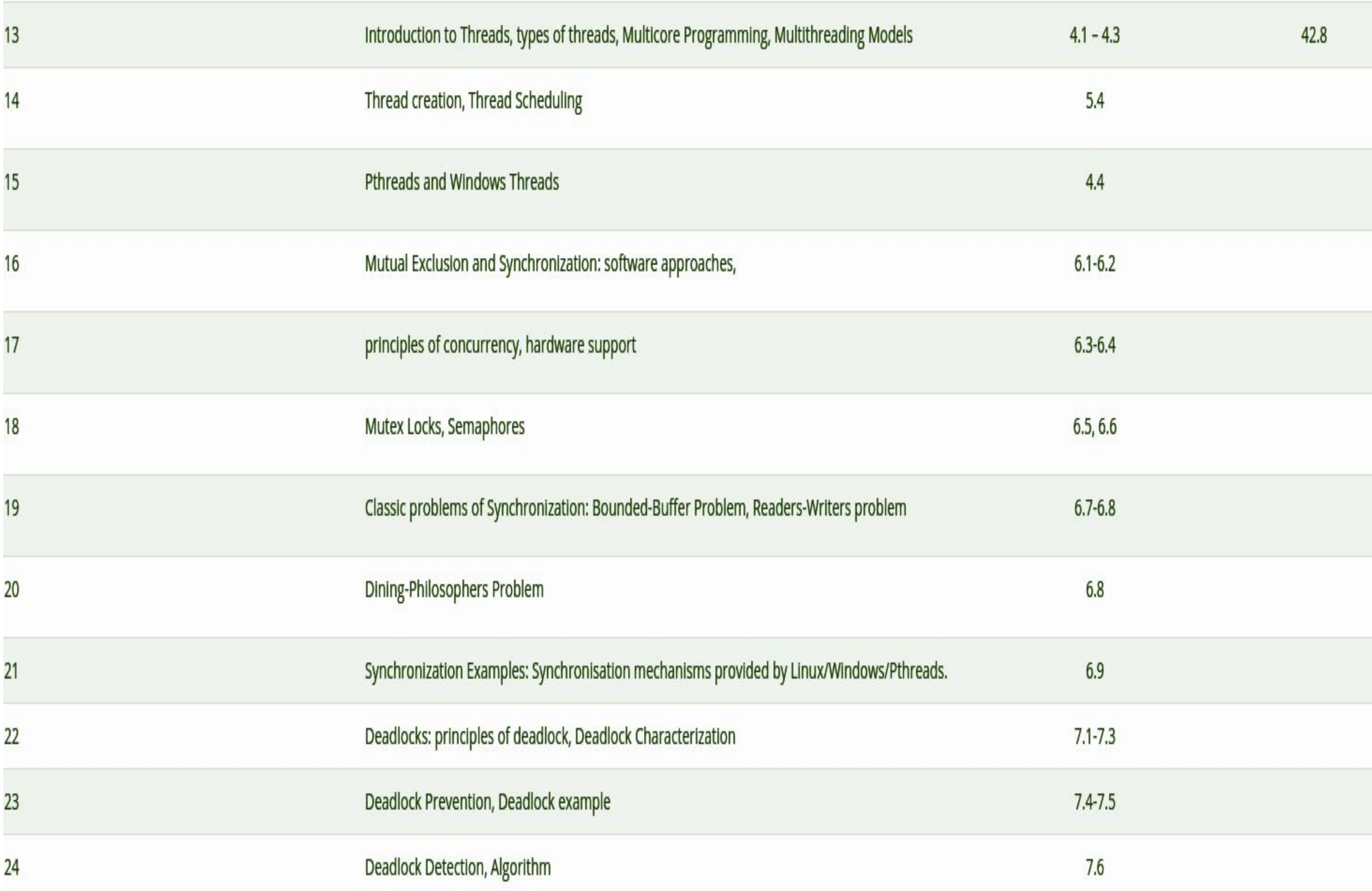
OPERATING SYSTEMS

Course Syllabus => Unit 1



Class No.	Chapter Title / Reference Literature	Topics to be covered	% of Portions covered	
			Reference chapter	Cumulative
1		Introduction: What Operating Systems Do, Computer-System Organization	1.1, 1.2	21.4
2		Computer-System Architecture, Operating-System Structure & Operations	1.3,1.4,1.5	
3		Kernel Data Structures, Computing Environments	1.10, 1.11	
4		Operating-System Services, Operating-System Design and Implementation	2.1, 2.6	
5		Process concept: Process in memory, Process State, Process Control Block, Context switch, Process Creation & Termination,	3.1 - 3.3	
6		CPU Scheduling - Preemptive and Non-Preemptive, Scheduling Criteria, FIFO Algrorithm	5.1-5.2	
7		Scheduling Algorithms:SJF, Round-Robin and Priority Scheduling	5.3	
8		Multi-Level Queue, Multi-Level Feedback Queue	5.3	
9		Multiprocessor and Real Time Scheduling	5.5, 5.6	
10		Case Study: Linux/ Windows Scheduling Policies.	5.7	
11		Inter Process Communication - Shared Memory, Messages	3.4	
12.		Named and unnamed pipes (+Review)	3.6.3	

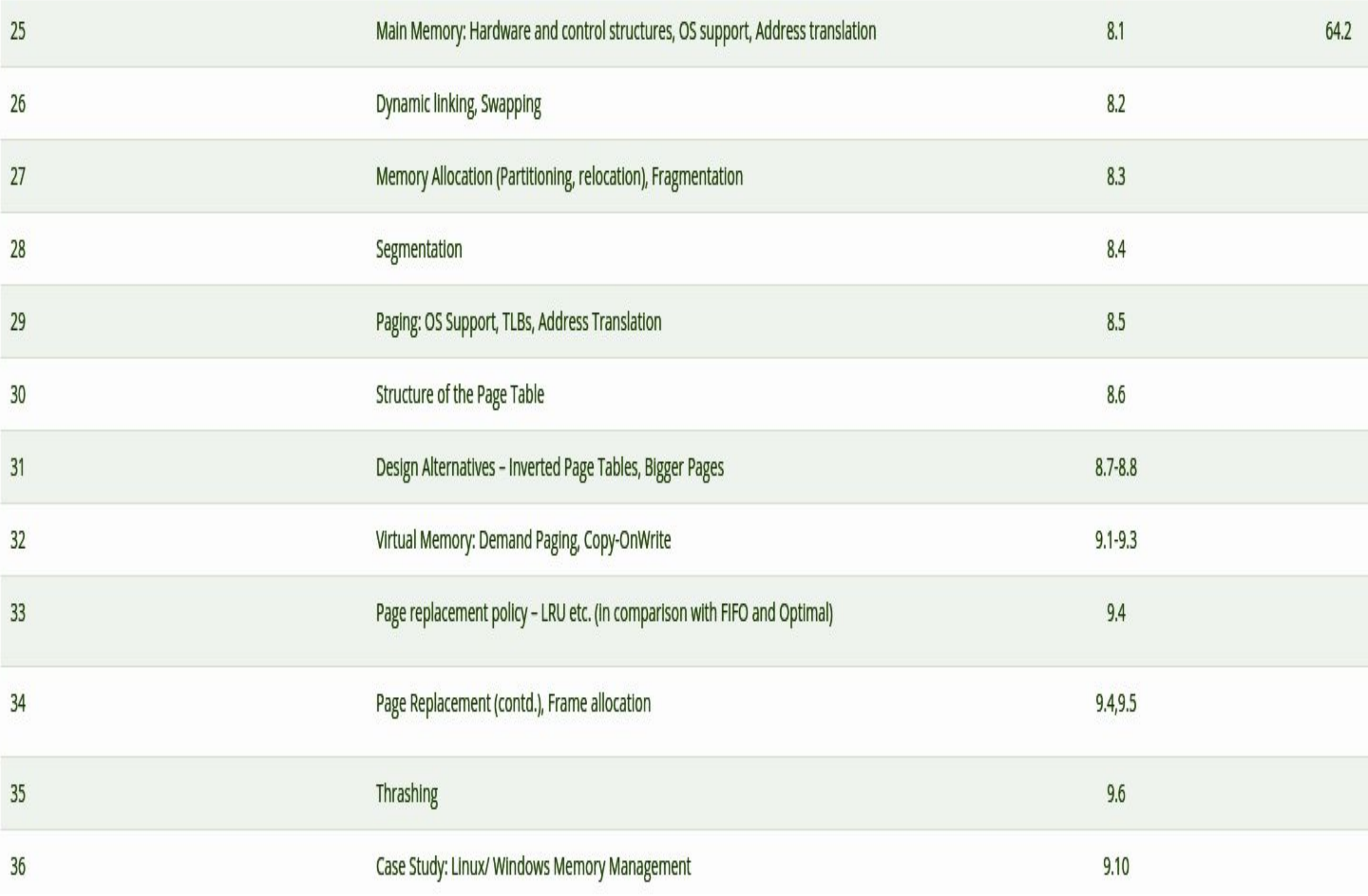
Course Syllabus => Unit 2



13	Introduction to Threads, types of threads, Multicore Programming, Multithreading Models	4.1 - 4.3	42.8
14	Thread creation, Thread Scheduling	5.4	
15	Pthreads and Windows Threads	4.4	
16	Mutual Exclusion and Synchronization: software approaches,	6.1-6.2	
17	principles of concurrency, hardware support	6.3-6.4	
18	Mutex Locks, Semaphores	6.5, 6.6	
19	Classic problems of Synchronization: Bounded-Buffer Problem, Readers-Writers problem	6.7-6.8	
20	Dining-Philosophers Problem	6.8	
21	Synchronization Examples: Synchronisation mechanisms provided by Linux/Windows/Pthreads.	6.9	
22	Deadlocks: principles of deadlock, Deadlock Characterization	7.1-7.3	
23	Deadlock Prevention, Deadlock example	7.4-7.5	
24	Deadlock Detection, Algorithm	7.6	

OPERATING SYSTEMS

Course Syllabus => Unit 3



25	Main Memory: Hardware and control structures, OS support, Address translation	8.1	64.2
26	Dynamic linking, Swapping	8.2	
27	Memory Allocation (Partitioning, relocation), Fragmentation	8.3	
28	Segmentation	8.4	
29	Paging: OS Support, TLBs, Address Translation	8.5	
30	Structure of the Page Table	8.6	
31	Design Alternatives - Inverted Page Tables, Bigger Pages	8.7-8.8	
32	Virtual Memory: Demand Paging, Copy-OnWrite	9.1-9.3	
33	Page replacement policy - LRU etc. (In comparison with FIFO and Optimal)	9.4	
34	Page Replacement (contd.), Frame allocation	9.4,9.5	
35	Thrashing	9.6	
36	Case Study: Linux/ Windows Memory Management	9.10	

Facts, Concepts, True / False, Fill in the Blanks

- A dual-core system requires each core has its own cache memory => **False**
- The operating system kernel consists of all system and application programs in a computer => **False**
- A **timer** can be used to prevent a user program from never returning control to the operating system.
- **Mainframe** operating systems are designed primarily to maximize resource utilization.
- Embedded computers typically run on a **realtime** operating system.
- Two important design issues for cache memory are **Size** and **Replacement** Policy
- If a program terminates abnormally, a dump of memory may be examined by a **debugger** to determine the cause of the problem
- A message-passing model is **easier to implement than a shared memory for intercomputer communication**
- The Windows CreateProcess() system call creates a new process. What is the equivalent system call in UNIX => **fork ( )**

Facts, Concepts, True / False, Fill in the Blanks

- **Modules** allow operating system services to be loaded dynamically

- When a child process is created, following is a possibility in terms of the execution or address space of the child process

    - The child process runs concurrently with the parent.
    - The child process has a new program loaded into it.
    - The child is a duplicate of the parent.
    - All the above

- The **stack** of a process contains temporary data such as function parameters, return addresses, and local variables.

- **shell** is the interface between the user and the operating system

- **system call** => A program request a service from an operating system's kernel

- **Data register** => Used in micro computer to temporarily store data being transmitted to/from an device

- **Address Register** => Portion of computer memory that keeps track of location in memory

Facts, Concepts, True / False, Fill in the Blanks

- **PC (Program Counter) =>** A register in a computer processor that contains the address (location) of the instruction being executed at the current time.

- **PSW (Program Status Word) =>** Controls the order in which instructions are fed to the processor, and indicates the status of the system in relation to the currently running program

- **wait for graph - used for deadlock detection for single instance of resource**

- **Banker's Safety Algorithm => used for Deadlock Avoidance**

- **Banker's Resource Request Algorithm => used for Deadlock Detection for Multiple instances of resources**

OS Exercise

Find Rmax, Need, check for safety - using Banker’s Algorithm

RMax		
A	B	C
?	?	?

Available=>Rmax-Allocated		
A	B	C
1	5	2

Process	Allocation			Max		
	A	B	C	A	B	C
P0	0	0	1	0	0	1
P1	1	0	0	1	7	5
P2	1	3	5	2	3	5
P3	0	6	3	0	6	5
Total	?	?	?	?	?	?

OS Exercise

Find Available, Need, check for safety - using Banker’s Algorithm

RMax		
A	B	C
4	2	5

Available=>Rmax-Allocated		
A	B	C
?	?	?

Process	Allocation			Max		
	A	B	C	A	B	C
P0	1	0	1	2	1	2
P1	0	0	1	3	2	4
P2	1	1	1	4	2	1
Total	?	?	?	?	?	?

Find Available, Maximum, check for safety - using Banker's Algorithm

RMax		
A	B	C
4	2	5

Available=>Rmax-Allocated		
A	B	C
?	?	?

Process	Allocation			Max		
	A	B	C	A	B	C
P0	1	0	1	?	?	?
P1	0	0	1	?	?	?
P2	1	1	1	?	?	?
Total	?	?	?	?	?	?

Need		
A	B	C
2	1	2
3	2	4
4	2	1
?	?	?

OPERATING SYSTEMS

OS Exercise

Req #	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
Page #	4	7	6	1	7	6	1	4	7	2

Fr #	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
1										
2										
3										
Flag										

Apply:

1. Optimal Page Replacement Policy
2. Most Recently Used (MRU) Page replacement policy
3. Least Recently Used (LRU) Page replacement policy

Working Set => {   }

Total Number of Page Requests = >

Total Number of Frames =>

Total Number of Page faults =>

% of Page Faults = >

% of Page Hits =>

Legend

Page Fault => PF

Page Hit => PH

OS Exercise

Req #	R1	R2	R3	R4	R5	R6
Page #	4	7	6	1	7	6

Apply:

1. Most Frequently Used (MFU) Page replacement policy
2. Least Frequently Used (LFU) Page replacement policy

Fr #	R1	FQ
1		
2		
3		
Flag		

Fr #	R2	FQ
1		
2		
3		
Flag		

Fr #	R3	FQ
1		
2		
3		
Flag		

Fr #	R4	FQ
1		
2		
3		
Flag		

Fr #	R5	FQ
1		
2		
3		
Flag		

Fr #	R6	FQ
1		
2		
3		
Flag		

Working Set => {   }

Total Number of Page Requests = >

Total Number of Frames =>

Total Number of Page faults =>

% of Page Faults = >

% of Page Hits =>

Legend
Page Fault => PF
Page Hit => PH
FQ => Frequency Count

OS Exercise

Req #	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20
Page #	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6

Fr #	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20
1																				
2																				
3																				
Flag																				

Working Set => {   }

Total Number of Page Requests = >

Total Number of Frames =>

Total Number of Page faults =>

% of Page Faults = >

% of Page Hits =>

Legend

Page Fault => PF

Page Hit => PH

Apply and Compare:

1. FIFO Page Replacement Policy
2. Most Recently Used (MRU) Page replacement policy
3. Least Recently Used (LRU) Page replacement policy

THANK YOU - Wishing you the Very Best for the upcoming ISA - 1

Nitin V Pujari
Faculty, Computer Science
Dean, IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com