



Baddies

22/02/2026

MULTI PROGRAMMING SYSTEM

padichalo?

TEAM

NandhaKishore

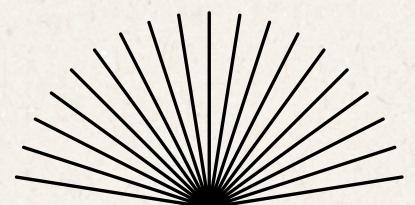
Shaza Jabbar

Vishnu L

Sreedeep Pradeep

Abhishek Sukumaran

Shad C T



Agenda

01	Overview
02	Process states in multiprogramming
03	Multiprogramming OS: Additional features
04	Process Control Block
05	Memory management in Multi Programming
06	Advantage and disadvantage

System	Kitchen Analogy (The "Vibe")	How it Handles the CPU
Uniprogramming (Old School)	One chef, one dish. He puts the bread in the toaster and stares at it until it pops.	CPU is idle during I/O. Very inefficient.
Multiprogramming (Our Topic)	While the bread is toasting, the chef switches to chopping onions for another dish.	Goal: CPU Efficiency. Switches jobs when one is waiting for I/O.
Multitasking (Modern Desktop)	The chef chops one onion, then flips one pancake, then stirs one soup—switching so fast it looks like he's doing it all at once.	Goal: User Interaction. Switches based on a Time Slice (e.g., 10ms per job) regardless of I/O.
Multiprocessing (High-end PC)	You have two chefs (Two CPUs) working in the same kitchen.	Goal: Speed/Power. Two separate processors executing different instructions at the exact same time.

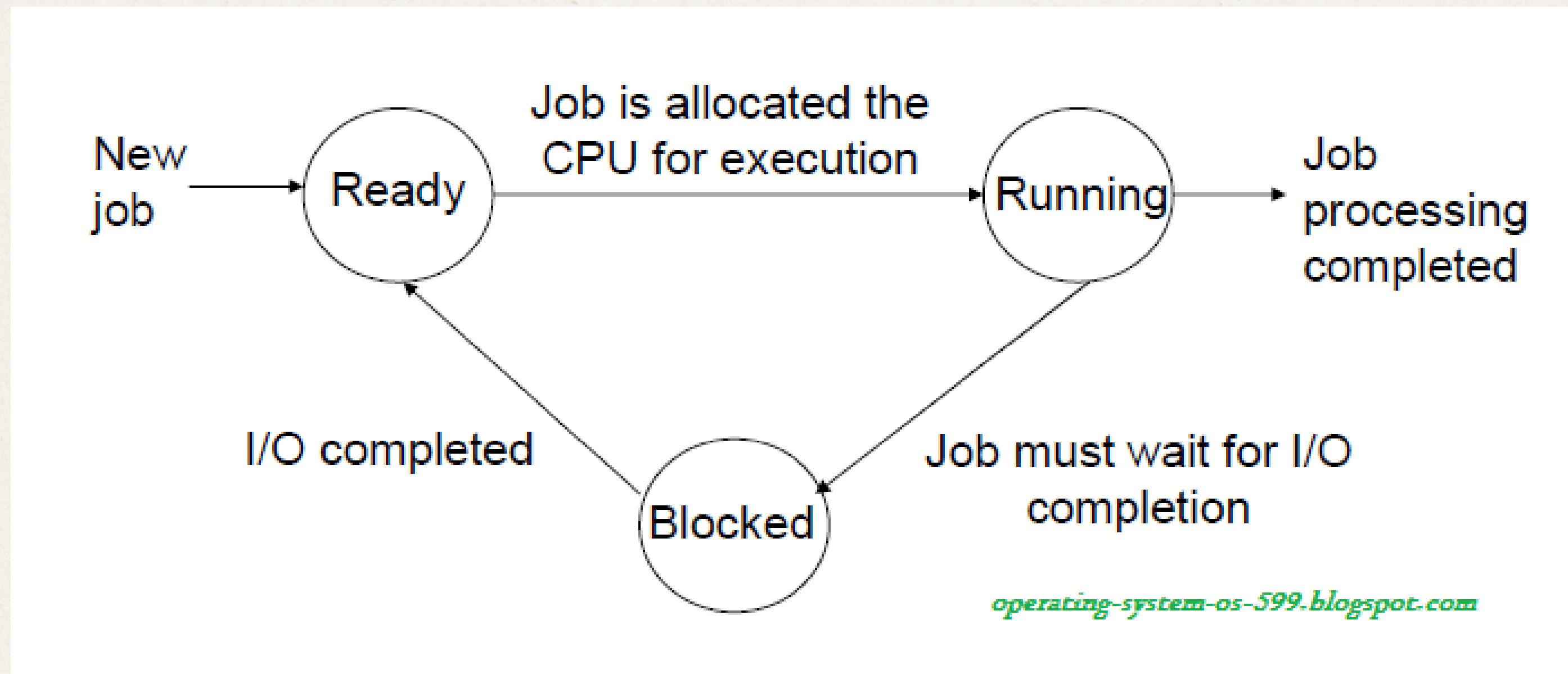


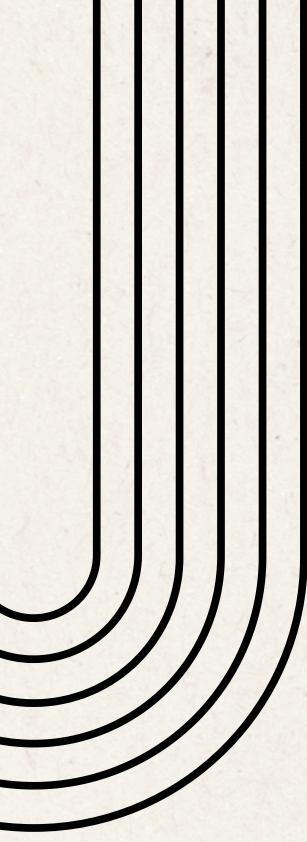
- 01** Multiprogramming is the interleaved execution of two or more different and independent programs by the same computer
- 02** The goal of multiprogramming- improve the system performance by exploiting the concurrency between the CPU and IO sub-system
- 03** While I/O sub-system is busy with one job, the CPU can execute another job
- 04** The area occupied by each program residing simultaneously in the memory is known as memory partition
- 05** The degree of multiprogramming: The number of programs actively competing for system resources
- 06** Higher degree of multiprogramming results in higher resource utilization

Overview

Multiprogramming improves system performance by allowing the CPU and I/O subsystem to work concurrently—while one job waits for I/O, the CPU executes another—through interleaved execution of multiple independent programs residing in separate memory partitions

Process states in multiprogramming





Multiprogramming OS: Additional features^{03/10}

- 01** Large memory
- 02** Memory protection: Prevent a program in one memory partition to access the partition of another program
- 03** Program status preservation: Program status should be preserved to restart a stopped program; use process control block
- 04** Proper job mix

Process Control Block

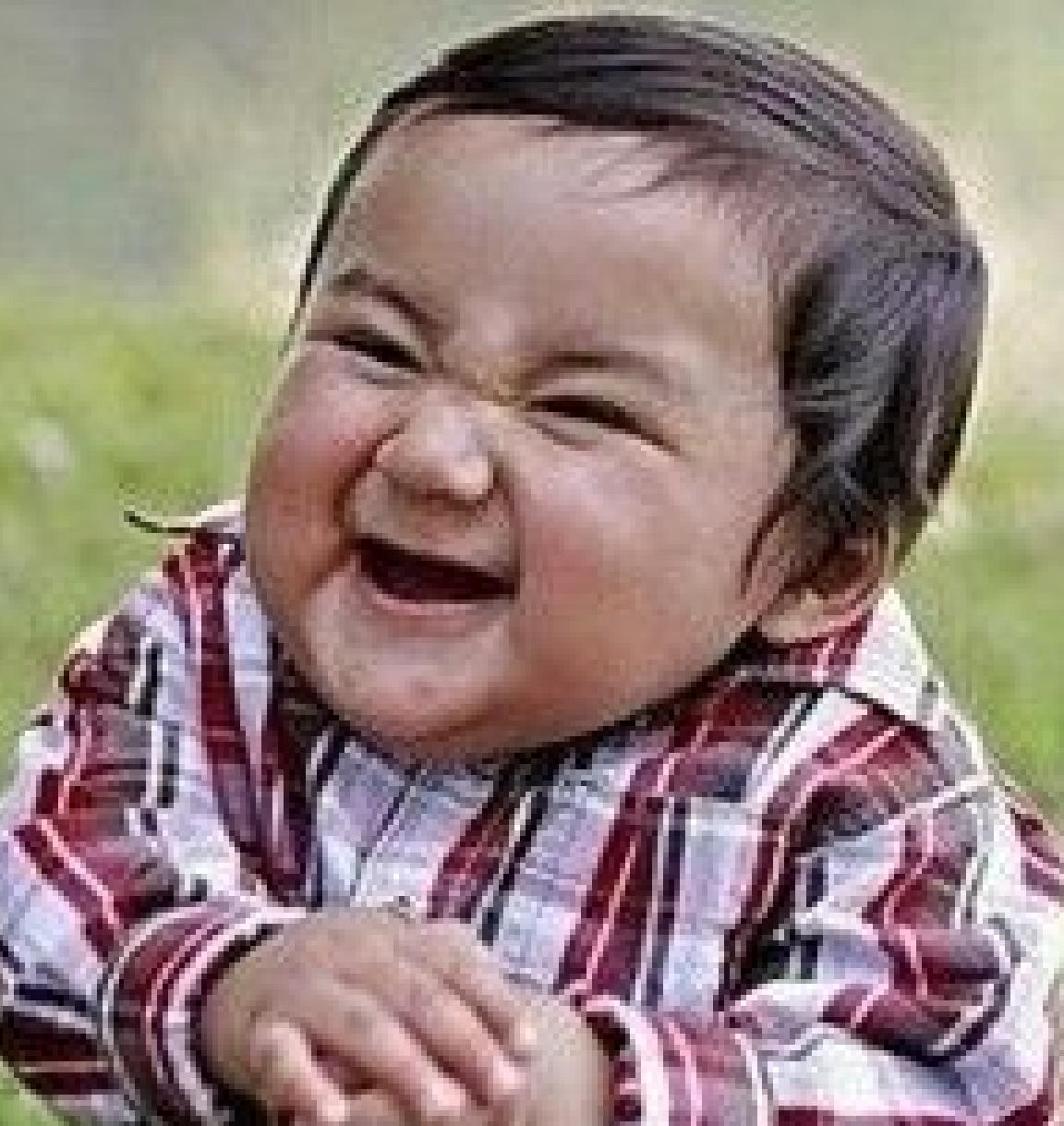
process identifier
process state
program counter
values of various CPU registers
accounting and scheduling information
I/O status information
:

Why do we use PCB? : for program status preservation

Process Control Block

03/10

QUIZ TIME



Process Control Block

03/10



Why do we use PCB?

A: for program status preservation

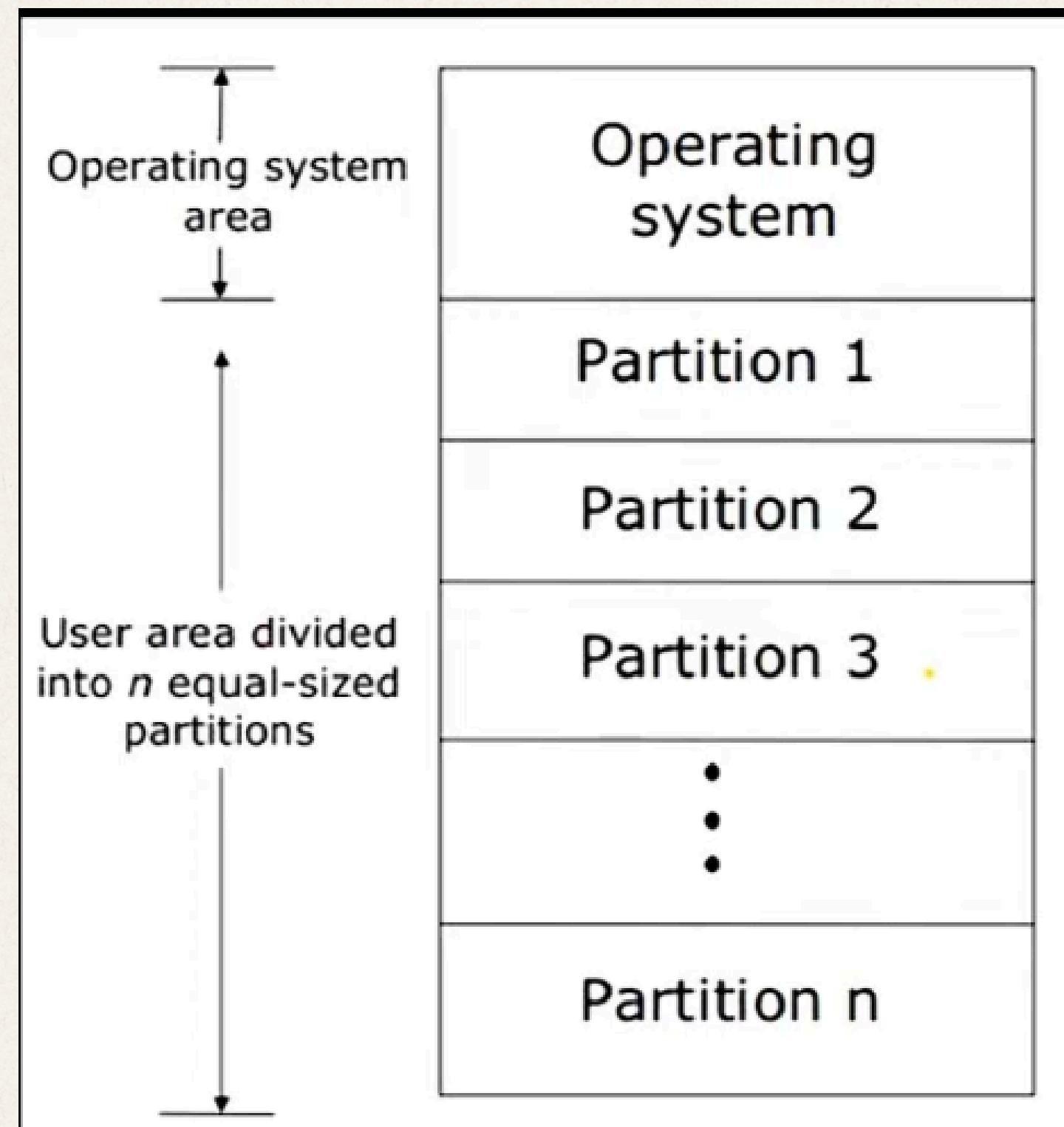
B: not for program status preservation

Memory management in Multi Programming

- **Fixed Partition (contiguous allocation technique)**
- **Variable Partition (non-contiguous allocation technique)**

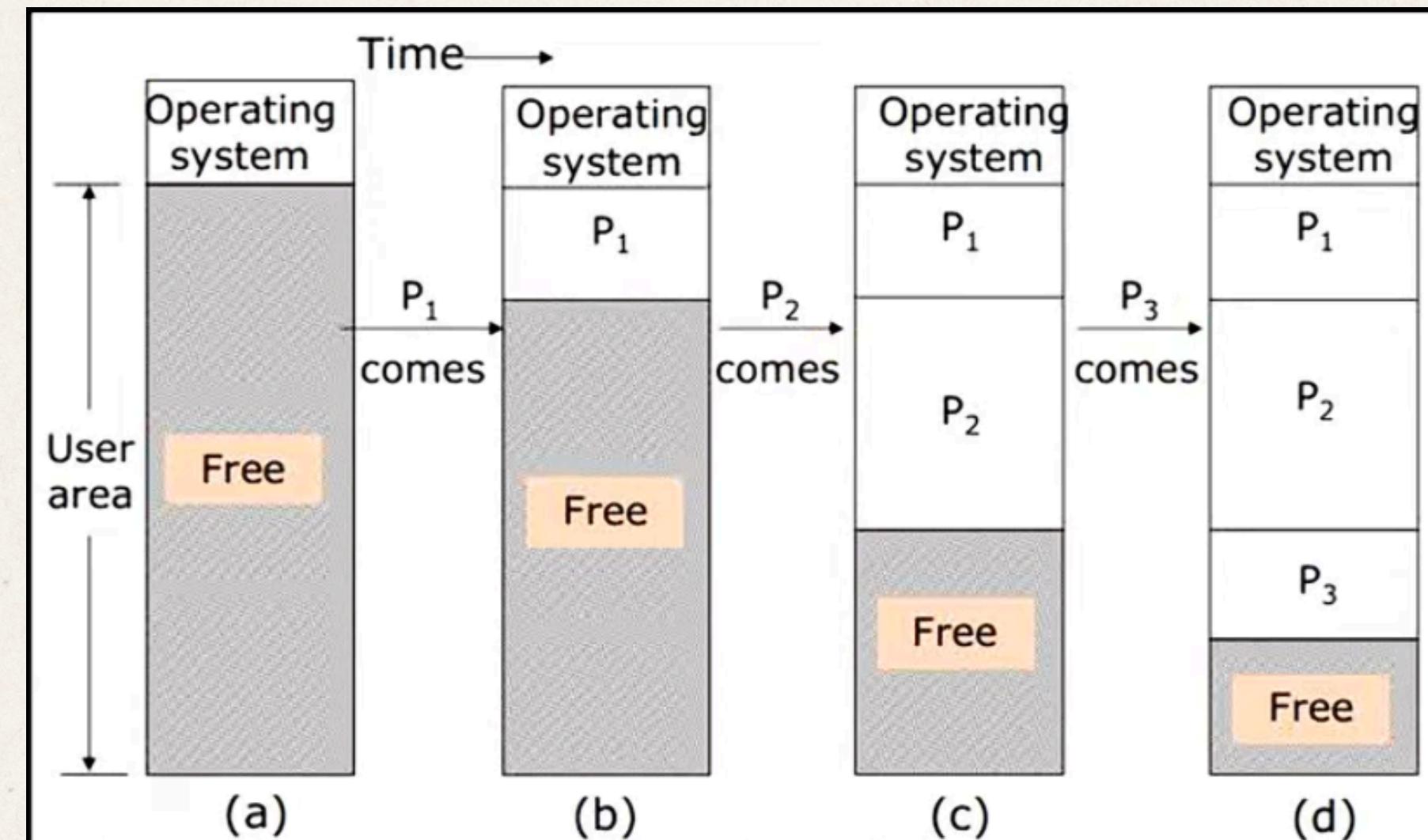
Fixed Partition (contiguous allocation technique)

- User area is divided into fixed number of partition.
- size of partition can be different
- memory is loaded based on space it has
- waste of memory in partition after use is : internal fragmentation - issue in fixed and simplest



Variable Partition (non-contiguous allocation technique)

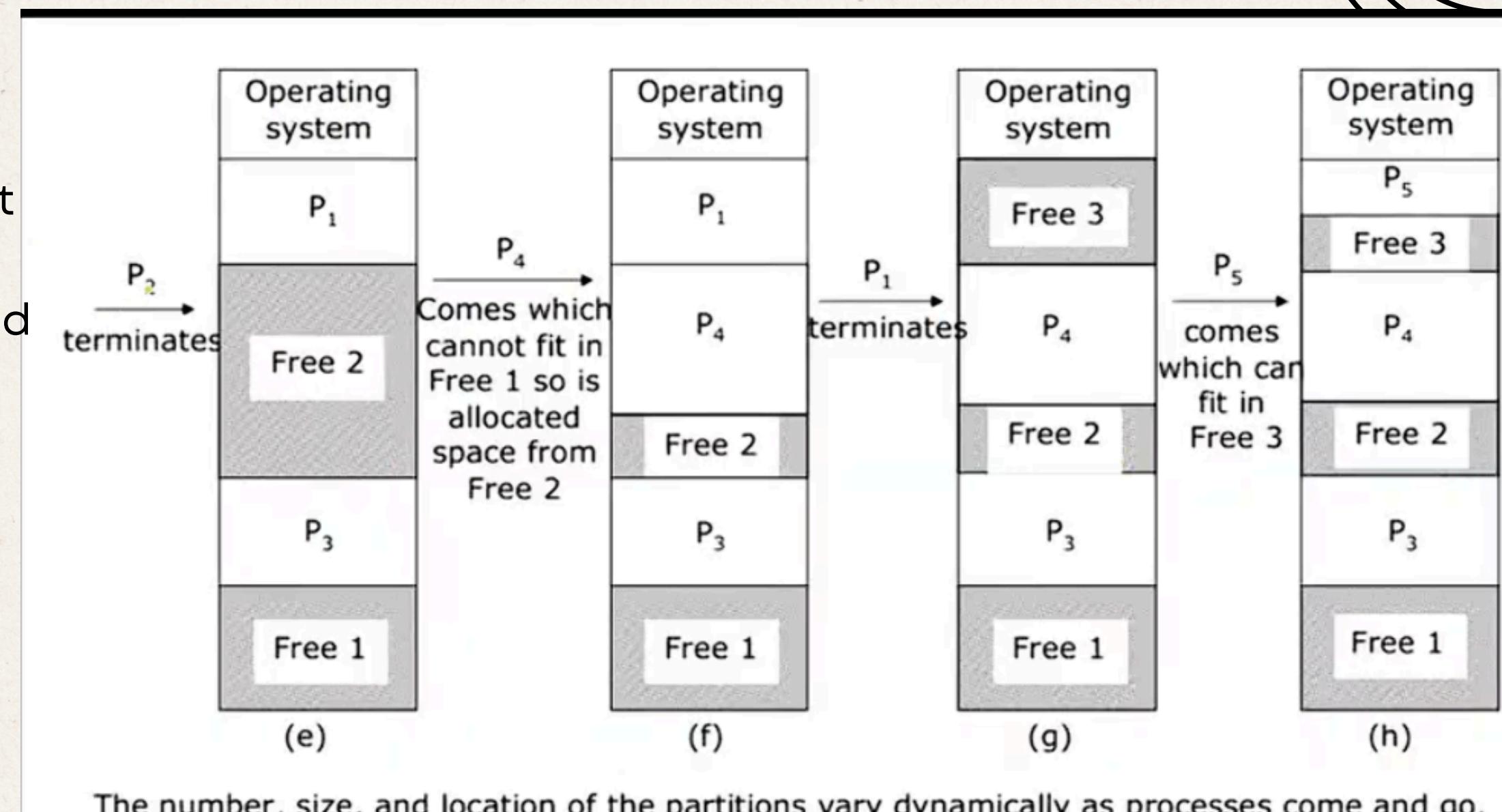
- Whenever a job comes in free memory available will be allocated



The number, size, and location of the partitions vary dynamically as processes come and go. (contd...)

Variable Partition (non-contiguous allocation technique)

- issue is that there are still free 2 ,3,1 left but we cannot access it due to p4 p5 p3
- for adding p4 , we have methods like : best fit, first fit.
- the free space here is called external fragment (wasted memory--)
- to avoid : use compassion technique . allocated are stacked. free also stacked





**Fixed Partition
(contiguous
allocation
technique)**



**Variable
Partition (non-
contiguous
allocation
technique)**

Advantages

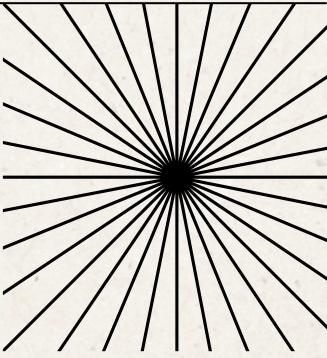
- CPU utilization is maximum
- Faster execution of short time jobs
- Resources are used efficiently
- Short response time

Disadvantages

- Complicated scheduling, hence difficult to program
- Sometimes long jobs have to wait long time

Our team

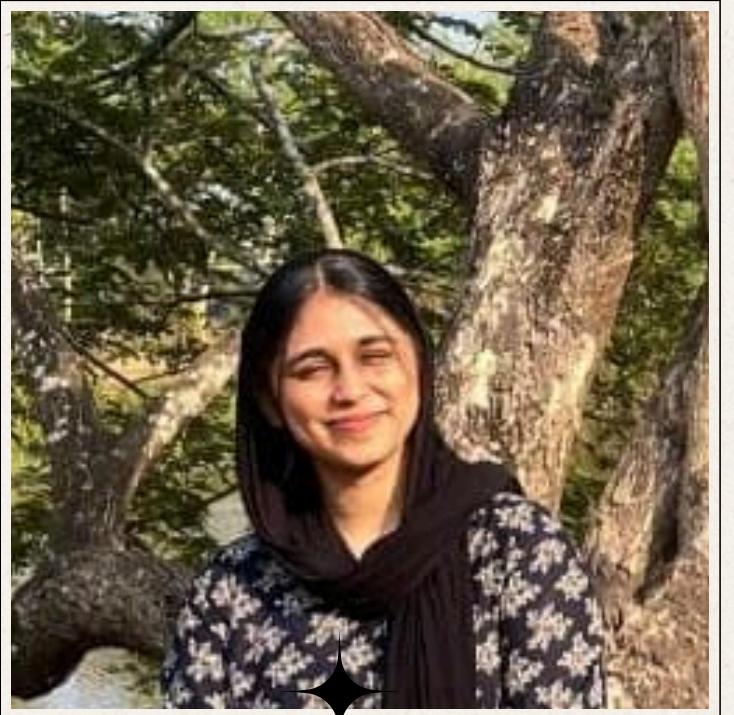
09/10



Sreedeepraj Sreechu



Shadu



Shaza rubber



Nandhu Chandhu



Vishnu Vichu



Abhishek Annan



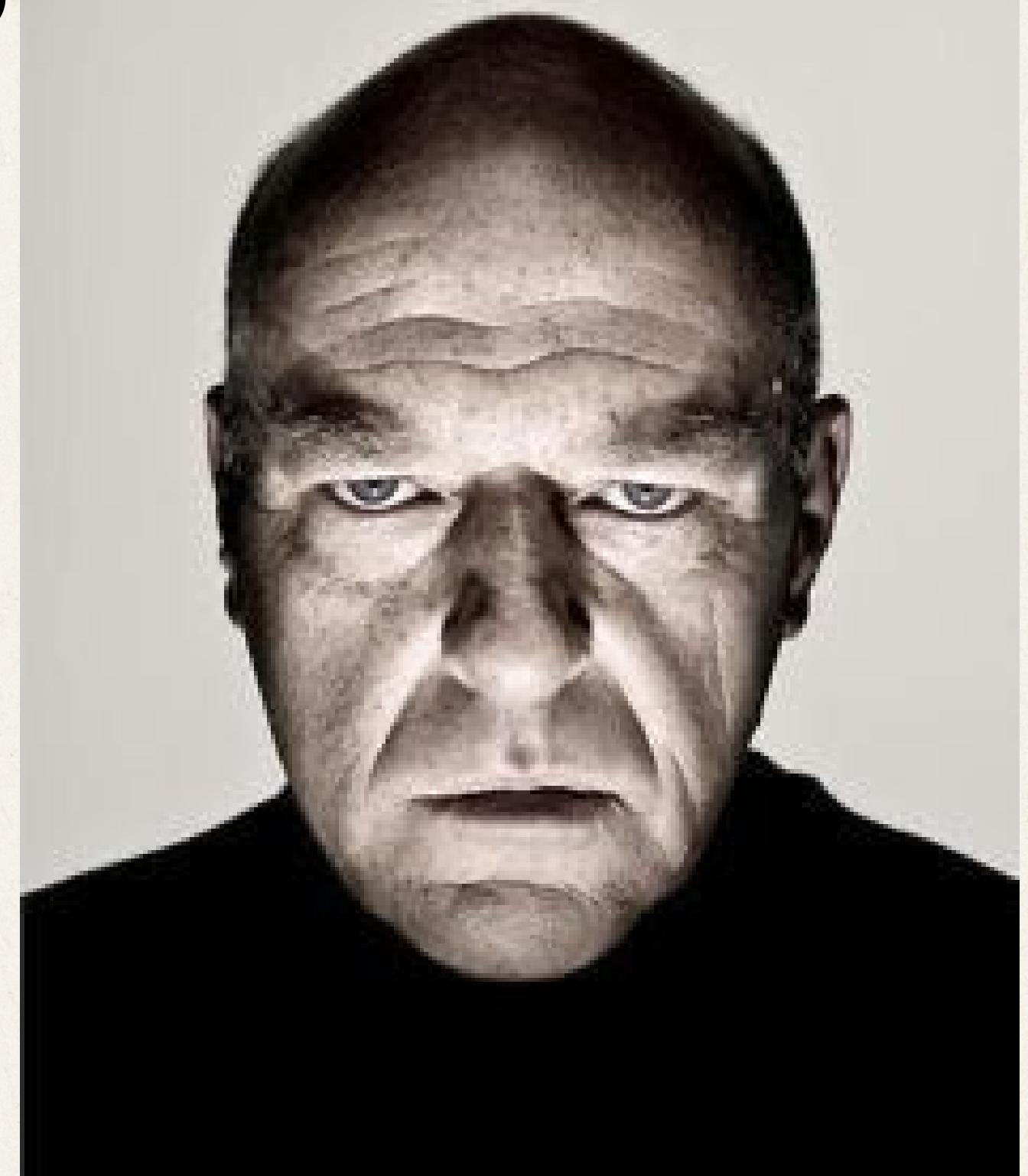
**THANK
YOU
GUYS!**

10/10



10/10

Any Doubts?



10/10

Thank you

