

## Operating System

**[Syllabus: BPSC CS:** Introduction, evolution, goals and components of OS. Types of OS Process management: Process states PCB, job and process scheduling. CPU scheduling algorithms, critical section problems and solutions. Semaphores, Inter-process communication techniques. Deadlock handling methods. Memory management techniques: Paging, segmentation and page replacement policies. Secondary storage management: Disk scheduling algorithms. File management: File system structure, organization, FCB, space allocation, tree structured file system. Protection and security: classification and handling techniques.

**NTRCA CS:** Definition and types OS, OS Structures: Processes: CPU scheduling: Process Synchronization: Deadlocks: Memory management: Virtual Memory: File Concept: File System Implementation

**NTRCA ICT:** Overview of operating systems, functionalities and characteristics of OS. Hardware concepts related to OS, CPU states, I/O channels, memory hierarchy, and microprogramming. The concept of a process, operations on processes, process states, concurrent processes, process control block, process context. UNIX process control and management, PCB, signals, forks and pipes. Interrupt processing, operating system organization, OS kernel FLIH, dispatcher. Job and processor scheduling, scheduling algorithms, process hierarchies. Problems of concurrent processes, critical sections, mutual exclusion, synchronization, deadlock. Mutual exclusion, process co-operation, producer and consumer processes. Semaphores: definition, init, wait, signal operations. Use of semaphores to implement mutex, process synchronization etc., implementation of semaphores. Critical regions, Conditional Critical Regions, Monitors, Ada Tasks. Interprocess Communication (IPC), Message Passing, Direct and Indirect, Deadlock: prevention, detection, avoidance, banker's algorithm. Memory organization and management, storage allocation. Virtual memory concepts, paging and segmentation, address mapping. Virtual storage management, page replacement strategies. File organization: blocking and buffering, file descriptor, directory structure, File and Directory structures, blocks and fragments, directory tree, inodes, file descriptors, UNIX file structure.]

**প্রশ্ন-১. Operating System এর মৌলিক ধারনা**

অপারেটিং সিস্টেম (অপারেটিং সিস্টেম) কাকে বলে? [Jnr. Instructor-2014, Different Ministry, BPSC, Instructor(Computer)-2018, MoE, BPSC, Assis. Pro - 2012, PSC, 38<sup>th</sup> BCS]

উত্তর: Operating System (অপারেটিং সিস্টেম): অপারেটিং সিস্টেম হলো এক ধরণের প্রোগ্রাম (Program) যা কম্পিউটার হার্ডওয়ার এবং User (ব্যবহারকারী) এর মধ্যে সময়স্থান সাঝে করে।

অন্যভাবে বলা যায় যে, অপারেটিং সিস্টেম হলো একধরণের প্রোগ্রাম (Program) বা একজন প্রোগ্রামের সমষ্টি, যা কম্পিউটারের সকল ক্রিয়াকলাপ (Operation) সম্পাদন ও নিয়ন্ত্রণ করে।

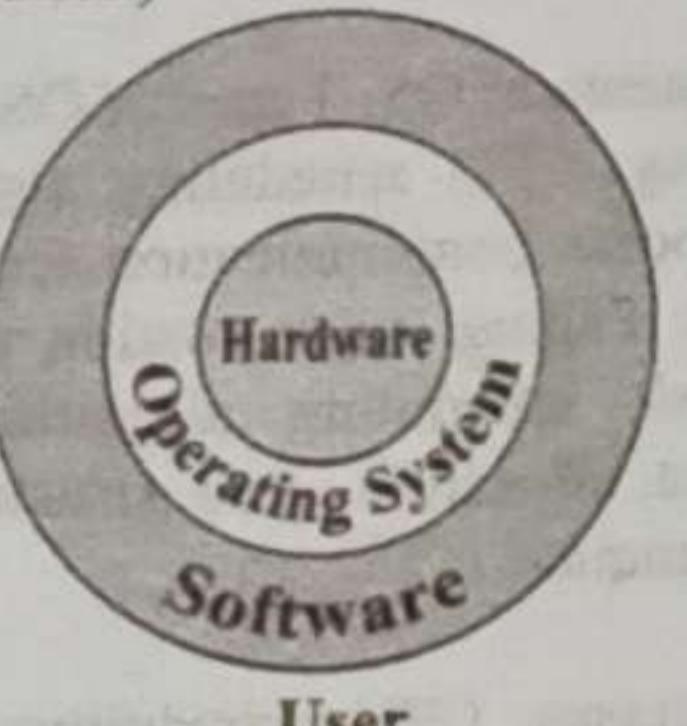


Figure: Operating System (অপারেটিং সিস্টেম)

### প্রশ্ন-২. Why do we need operating system? [ 38<sup>th</sup> BCS ]

আমাদের অপারেটিং সিস্টেম প্রয়োজন। কারণ:

1. ইউজার এবং হার্ডওয়ারের মধ্যে সংযোগ স্থাপন করার জন্য।
2. হার্ডওয়ারকে কার্যকরীভুলভাবে অপারেটিং সিস্টেম সরবকার।
3. তাছাড়া বিভিন্ন কাজ করে থাকে যেমন:

  - i. প্রসেস ম্যানেজমেন্ট (Process Management)
  - ii. মেইন মেমোরি ম্যানেজমেন্ট (Main Memory Management)
  - iii. ফাইল ম্যানেজমেন্ট (File Management)
  - iv. ইনপুট/আউটপুট ম্যানেজমেন্ট (Input/Output Management)
  - v. যোগাযোগ ব্যবহাপনা (Communication Management)

### প্রশ্ন-৩. Operating System এর বিবরণ আলোচনা করুন।

উত্তর: কম্পিউটারের আবকারের সূচনালয়ে কম্পিউটারের সিস্টেমে কোন অপারেটিং সিস্টেম ছিলোনা। কম্পিউটারের ব্যবহারকারীগণ মেশিন ল্যাঙ্গুেজের প্রয়োগের মাধ্যমে কম্পিউটারের রিসোর্সকে Access বা সম্পাদন করতো। তখন কম্পিউটারের অপারেটরগণ ম্যানুয়ালি বিভিন্ন

প্রজন্ম (Generation)	প্রযুক্তি (Technology)	(Software/Operating System)	(Computer/Machine)	বৈশিষ্ট্য (Features/Characteristics)
প্রথম প্রজন্ম (১৯৪৫-৫৮)	ভ্যাকুম টিউবও <sup>১</sup> বিলে মেমোরি, PC ব্যারা নিয়ন্ত্রিত ও নির্দিষ্ট Accumulator এর Arithmetic	Machine/Assemble Language, একক ব্যবহার পক্ষত।	ENIAC, IBM-701, Princeton IAS.	Computer Memory এ Manually Program Input করার পর Execute করা হতো।

কম্পিউটারে Send (Enter) করে প্রোগ্রামকে Initiate (অঙ্গ সময় অপচয় হতো এবং সিপিইউ (CPU) অসমতাবে অঙ্গ কম্পিউটারে প্রক্রিয়ালয়ের নিরলস প্রচেষ্টায় ধাপে ধাপে অপারেটিং সিস্টেম তলো বর্তমানে অপারেটিং সিস্টেম তলো থেকে আমরা যেসকল সুযোগ সৃষ্টি পাই তা ধাপে ধাপে উন্নতি হতে সময় লেগেছে আয় ৫০ থেকে ১৫ বছর।

সর্বপ্রথম ১৯৫০ সালে USA এর General Motor Research Laboratory কর্তৃক IBM Corporation এর IBM-701 Mainframe কম্পিউটারে ব্যবহার করা হচ্ছে। কম্পিউটারের বিভিন্ন জব (Job) এর মধ্যে ট্রানজিস্টরের জন্য অপারেটিং সিস্টেম ডেভেলপ করা হয় এবং তখন থেকেই ব্যাচ প্রসেস হচ্ছে, অর্থাৎ প্রোগ্রামকে বিভিন্ন ব্যাচে (Batch) Group এর Execute বা সম্পাদন করা হতো।

১৯৬০ সালের দিকে অপারেটিং সিস্টেম অধিকতর Sophisticated ভাবে তৈরি করা হয় এবং ক্রমে ক্রমে মাল্টিপ্রোসারিস, টাইম শেল, মাল্টি প্রসেসিং অপারেটিং সিস্টেম তৈরি করা হলো এবং যিলে টাইম অপারেটিং সিস্টেম ও বাজারে আসতে চৰুত করলো। শাঠের দলকে প্রেসিলিক এবং স্বতরের দশকের ক্ষেত্রে নিকে General Purpose অপারেটিং সিস্টেম বাজারে আসতে থাকে, যা ছিলো মাল্টি মেইন সিস্টেম। আবার এগুলো কিছু কিছু অপারেটিং সিস্টেম Simultaneously Batch Processing, Time Sharing Real Time Processing এবং Multiprocessing Suppon করতো। যেমন: IBM-360 Family Group এর অপারেটিং সিস্টেম।

১৯৭৬ সালের দিকে মাইক্রোকম্পিউটারে অপারেটিং সিস্টেমের ব্যবহার শুরু হয়। যার নাম CP/M(Control/Program for Microprocessor)। কিন্তু এসকল অপারেটিং সিস্টেমে Automatic Job to Job transition ছিলোনা, Job Load ও Unload করে যাতে Operation করা হতো। নতুন কোন Job রান করে হলে পূর্বের Job এর ডেটা সমূহ মেমোরি থেকে ডিলেট করা দাগতে, ইনপুট/আউটপুট ডিভাইসের তুলনায় এর Speed বেশি ছিলো বলে এই Idle time বৃদ্ধি পায় এবং এটিকে Efficient ও Economically ব্যবহার করা সুব্রত হতোনা।

ঠিক এই সকল কারণে প্রয়োবত্তি অপারেটিং সিস্টেম ধাপে ধাপে ডেভেলপ হতে থাকে। নিচে অপারেটিং সিস্টেম এবং কম্পিউটারের পর্যায়ক্রমিক ডেভেলপমেন্ট টেক্সিল বা বিবরণ দেখানো হলো:

Operating System				
প্রতীয় প্রজন্ম (১৯৫৫-৬৪)	Discrete Transistors এবং Core Memory, FP Arithmetic, I/O Processors.	High Level Language এবং ব্যবহার Compilers batch Processing Monitor এর সাথে।	IBM-7090, CDC=1604, UNIVAC, LARC.	Batch System এর ভুক্ত, CPU Execution time কমায়, যেমনঃ FMS (Fortal) Monitor System।
তৃতীয় প্রজন্ম (১৯৬৫-৭৪)	IC, (SSI/MSI) Micro, Programming, Pipeline Cache & Lookahead Processors.	Multiprogramming & Time Sharing Operating System, Multiuser Applications.	PDP-8, CDC-6600, TI-ASC, IBM-360/370.	Multiprogramming System এর ব্যবহার, ২য় প্রজন্মের তুলনায় Program Execution এর সময় কম লাগে।
চতুর্থ প্রজন্ম (১৯৭৫-৯০)	LSI, VLSI, & Semiconductor Memory, Multiprocessor, Vector Super Computer, Multicomputer.	Multiprocessors Operating System, Language, Compilers, Parallel Processing.	VAX-9000, CRAY-X-MP, IBM-3090, BBn-TC-2000.	Personal Computer, Network, Distributed & Operating System, GUI, DOS, Windows.
পঞ্চম প্রজন্ম (১৯৯০- বর্তমান)	ULSI/VHSIC Processor, Memory & Switches, High Density Packaging, Scalable Architecture.	Massively (ব্যাপক) Parallel Processing, Heterogeneous (বিভিন্নধর্মী) Processing.	Fujitsu VPP-500, CRAY/MPP, TMC/CM-5, Intel Paragon, MIT-J-Machine, Caltech Mosaic.	GUI, Automatically Translators, User এর কাজ করানো, LINUX, Windows-2000, NT, XP- এর ব্যবহার।

### প্রশ্ন-৪. কার্নেল (Kernel) কী? কার্নেলের কার্যাবলী তলো লিখুন?

উত্তর: কার্নেল (Kernel): কার্নেল হলো অপারেটিং সিস্টেমের তরুণ ভূপূর্ণ অংশ। কার্নেল বিভিন্ন প্রোগ্রাম সমূহকে একেকটিকে একেকে সময় চলাতে দেয়। অর্থাৎ প্রসেসর ব্যবহাপনায় প্রত্যেককে সময় ভাগ করে দেয়। এছাড়া কার্নেল অনেক তলো প্রোগ্রাম এর মধ্যে সময়স্থান সাধন করে।

কার্নেলের কার্যাবলি সমূহ নিচে তুলে ধরা হলো:

- a) কার্নেল প্রোগ্রাম সমূহকে মেইন মেমোরিতে Load বা সরবরাহ করে।
- b) কার্নেল প্রসেসর ব্যবহাপনায় বিভিন্ন প্রোগ্রাম সমূহকে সময় ভাগ করে দেয়। যার ফলে প্রসেসর Time Slice অনুযায়ী Multiuser & Multitasking পরিবেশে কাজ করতে পারে।
- c) কার্নেল অনেক তলো প্রোগ্রামের মধ্যে সময়স্থান করে।
- d) মেমোরি ম্যানেজমেন্টের মাধ্যমে অপারেটিং সিস্টেম প্রত্যেক Job এর জন্য Space নির্দিষ্ট করে দেয়। ফলে সিস্টেম Crash করার সম্ভাবনা কমে যায়।
- e) মাল্টিপ্রোসারিস এর ব্যবহার হয় বলে সিস্টেমের Throughput বৃদ্ধি পায়।
- f) প্রত্যেকটি হার্ডওয়ারের জন্য সঠিক সফটওয়্যার ড্রাইভিং ব্যবহার করা কার্নেলের অন্যতম কাজ।
- g) কার্নেলের সাহায্যে প্রোগ্রামের ত্রুটি বা সমস্যা সমাপ্ত ও সংশোধন করা যায়।

h) মাল্টিপ্রোসারিসের প্রয়োজন অনুযায়ী ডিভাইস সমূহ সরান্ব করে।

i) বাফারিং (Buffering), ক্যাচিং (Caching) ও স্পুলিং (Spooling) কার্যাবলী সম্পর্ক করে।

j) সেকেন্ডারি মেমোরি ও ইনপুট/আউটপুট ব্যবহাপনার ফলে ইনপুট/আউটপুট পোর্ট, DMA Channel, Interrupt Request এর কাজ সমূহ সঠিকভাবে সম্পর্ক করে।

### প্রশ্ন-৫. অপারেটিং সিস্টেমের কার্যাবলী তলো বিস্তারিত আলোচনা করুন।

[Data entry supervisor – 2019 BPSC, Instructor ME WOE - 2018 BPSC, ক্ষেত্র যোগাযোগ ও প্রযুক্তি বিভাগ-২০১৯-ভাটা এন্ড্রিয়াসেরজাইজার, [ক্ষেত্র ও যোগাযোগ প্রযুক্তি অধিদলের ২০১৭-নেটওর্ক ইঞ্জিনিয়ার ও সহকারী প্রোগ্রামার (নল-ক্যাটার)]]

উত্তর: অপারেটিং সিস্টেম ব্যাক্তিত বর্তমান যুগে কম্পিউটারের ব্যবহার করা প্রায় অসম্ভব। কম্পিউটার হার্ডওয়ারের ও কম্পিউটার ব্যবহারকারীর মধ্যে সময়স্থান করাই হচ্ছে অপারেটিং সিস্টেমের প্রধান কাজ। এছাড়া ও অপারেটিং সিস্টেম বহুবিধ কাজ সম্পর্ক করে থাকে। নিচে সেগুলো তুলে ধরা হলো:

#### (ক) প্রসেস ম্যানেজমেন্ট (Process Management):

- a) User ও সিস্টেম প্রসেস তৈরি করা এবং মুছে (Delete) কেলা।
- b) প্রসেস সিনক্রেনাইজ (Process Synchronization) করা।
- c) ডেভেলক হ্যান্ডলিং (Deadlock handling) করা।
- d) প্রসেস Suspension (বিলড) এবং Resumption (পুনরাবৃত্ত) করা।
- e) প্রসেস কমিউনিকেট (Process Communicate) করা।

(g) মেইন মেমোরি ম্যানেজমেন্ট (Main Memory Management):

- a) মেইন মেমোরি ও অন্যান্য স্টোরেজকে সিস্টেম প্রোগ্রাম এবং চেটা অশে Allocate (বরাদ্দ) করা।
- b) মেমোরির কোন অশে ব্যবহার হচ্ছে তার হিসাব রাখা।
- c) কোন প্রসেস গুলো মেমোরিতে Load (সরবরাহ) হচ্ছে এবং কোন Release (মৃত্তি) হচ্ছে তা ম্যানেজ করা।
- d) অবৈজ্ঞানিকভাবে Allocate & Deallocate করা।

(h) ফাইল ম্যানেজমেন্ট (File Management):

- a) কোন ফাইল তৈরি করা ও ডিলেট করা।
- b) ডিলেটের পুরো তৈরি করা ও ডিলেট করা।
- c) ফাইল ও ডিলেটের মধ্যে Manipulate (সুনিশ্চিতভাবে) ব্যবহার করার জন্য Primitives Support দেয়।
- d) সেকেন্ডারি স্টোরেজে ফাইলকে মাপিং করা।
- e) ব্যাকআপ ফাইল তৈরি করা।

(i) ইনপুট/আউটপুট ম্যানেজমেন্ট (Input/Output Management):

- a) বাফারিং (Buffering), ক্যাচিং (Caching) ও স্পুলিং (Spooling) কার্যান্বয় সাধন করা।
- b) ইনপুট ও আউটপুট এর সমর্থন সাধন করা।
- c) I/O ডিটাইনের মাধ্যমে একাধিক প্রোগ্রাম এর মধ্যে সমর্থন সাধন ও নিয়ন্ত্রণ করা।

(ii) সেকেন্ডারি স্টোরেজ ম্যানেজমেন্ট (Secondary Storage Management):

- a) ফিল্পেস ম্যানেজমেন্ট করা।
- b) স্টোরেজ এলোকেশন করা।
- c) ডিক পিভিটলিং করা।

(j) মোগামোল ব্যবহার (Communication Management):

- a) Create, Delete & Communication Connection করা।
- b) মেসেজ আদান প্রদান করা।
- c) Status Information সেন্সের করা।
- d) Remote Devices সমূহকে Attach বা Detach করা।

(k) তথ্য ব্যবহার (Information Management):

- a) ফেপ আকারের ফাইল সমূহকে ম্যানেজ করা।
- b) ফাইলের ডিরেক্টোরিজ ম্যানেজ করা।
- c) ফাইলসহ রেকর্ডস সমূহকে ম্যানেজ ও প্রসেস করা।

উপরোক্ত ধরন ধরনের কাজ গুলো ছাড়াও অপারেটিং সিস্টেম নিম্নের কাজ গুলোও সম্পাদন করে দাকে: যেমন:

- a) হার্ডওয়ারের অবহান পর্ববেক্ষণ, কৃতি চিহ্নিকরণ ও সম্পোষণের ব্যবহার করে।
- b) মেইন মেমোরিতে প্রোগ্রাম সরবরাহ করা ও প্রিচালনা করা।
- c) অপারেটিং সিস্টেমের নির্বাচন, বিশ্বেষণ ও সম্পাদন করা।
- d) নিরাপত্তা ও সোপনীয়তা রক্ষার জন্য পাসওয়ার্ড ব্যবহারের সুবিধা প্রদান করা।
- e) কমান্ট ও ইন্ট্রুকশন গুলোকে ট্র্যান্সলেট করা।
- f) GUI এর ব্যবহাৰ করা।

- g) Job transition & Job Scheduling করা।
- h) চেটা ও ফাইল স্টোরেজকে ম্যানেজ করা।
- i) CPU কে বিভিন্ন টাকের জন্য Assign করা।
- j) কম্পিউটারকে Boot করা এবং অপারেশন ডক করা।

প্রশ্ন-৬. অপারেটিং সিস্টেম কীভাবে Hardware (হার্ডওয়ার) এবং User (ব্যবহারকারীকে) আলাদা বা স্থূল রাখে? চিহ্ন দিয়ে করুন।

উত্তর: সাধারণত একজন User বা ব্যবহারকারী অপারেটিং সিস্টেম Application Program এবং চেটা সরবরাহ করে থাকে এবং অপারেটিং সিস্টেম হার্ডওয়ারকে নিয়ে User (ব্যবহারকারীর) মধ্যে Execute (নির্বাচন) করে Output (ফলাফল) থালান করে। যার ক্ষেত্রে ব্যবহারকারী মনে করেন, তার দেওয়া Command (নির্দেশ) ও তিনিটে হার্ডওয়ার নির্ধারিত কাজটি Execute (নির্বাচন) করে। যি প্রক্রিয়কে অপারেটিং সিস্টেম কীভাবে হার্ডওয়ারকে নিয়ে কাজ করে থাকে যা নির্বাচন করে নেয় তা ব্যবহারকারীর নিকট অপোন্ট বা টাই হয় যায়। অর্থাৎ হার্ডওয়ার ও User (ব্যবহারকারীর) মধ্যে অপোন্ট Transparent বা বাজ পর্নার ন্যায় কাজ করে থাকে।

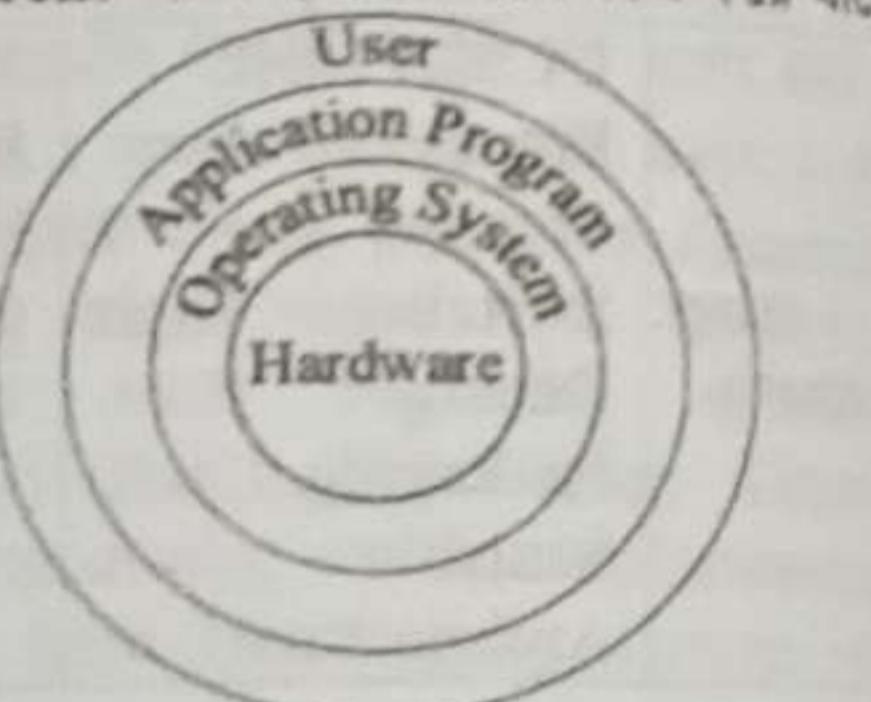


Figure: Operating System (অপারেটিং সিস্টেম)

প্রশ্ন-৭. Explain how hardware and operating system work together to control I/O devices.

[কৃতি অবস্থা অবলম্বন কৃতি প্রশিক্ষণ একাডেমি (সার্ট)-২০১১]

উত্তর: Operating System (অপারেটিং সিস্টেম): অপারেটিং সিস্টেম হলো এক ধরনের প্রোগ্রাম (Program) যা কম্পিউটার হার্ডওয়ার ও সিস্টেম করে।

কম্পিউটার সিস্টেমে প্রোগ্রাম বা চেটা স্ক্রিপ্ট প্রসেস করার জন্য অপারেটিং সিস্টেমের অর্থীভূত কন্ট্রোল প্রোগ্রামের একটি উচ্চতম পুরুষ ইনপুট/আউটপুট কন্ট্রোল প্রোগ্রাম (Input/Output Control Program)। ইহা কম্পিউটার হার্ডওয়ার তথ্য CPU এর সাথে বিভিন্ন ইনপুট/আউটপুট ডিভাইস গুলোর সময়স্থান সাধন করে এবং তথ্য অন্য অন্য প্রদানের কাজ পরিচালনা ও নিয়ন্ত্রণ করে।

প্রশ্ন-৮. Operating System এর উদ্দেশ্য কীটি ও কী কী?

Operating System এর উদ্দেশ্য গুলো লিখুন।

উত্তর: অপারেটিং সিস্টেমের দুই ধরণের উদ্দেশ্য রয়েছে। যথা:

(ক) Primary Goal (প্রদান উদ্দেশ্য):

- a) ব্যবহারকারীর প্রয়োগকে Execute (নির্বাচন) করা এবং ব্যবহারকারীদের সমস্যা গুলো সহজে সমাধান করা।
- b) কম্পিউটার সিস্টেমকে সুবিধাজনকভাবে ব্যবহার করা।

(খ) Secondary Goal (সহায়ক উদ্দেশ্য):

হার্ডওয়ারকে দক্ষতার সাথে ব্যবহার করা।

প্রশ্ন-৯. বিভিন্ন প্রকার অপারেটিং সিস্টেমের বৈশিষ্ট্য (Characteristics) লিখুন। [Jnr. Instructor-2014 Different Polytechnic BPSC]

- a) লিনার অপারেটিং সিস্টেমের বৈশিষ্ট্য সমূহ;
- b) ইয় ইনিক্স (UNIX) এর সাথে অতি সহজই সাম্পর্কপূর্ণ(Compatible);
- c) লিনারের উন্নত মানসম্পন্ন লিনার কার্নেল (Linux Kernel) রয়েছে।
- d) লিনারের সোর্স কোড বিনামূল্যে পাওয়া যায় বলে ইয় ইনিক্স ওয়ার্কস্টেশন এর বিকল হিসেবে ব্যবহৃত হচ্ছে।
- e) বৰ্তমানে লিনারের ভার্সনটি Free Software Foundation (FSF) এর প্রোটোকল।
- f) ইয় মাইক্রো এবং কনফিগুরে করা সহজ।
- g) লিনার হচ্ছে মাইক্রো ও মাইক্রোইনজার অপারেটিং সিস্টেম।
- h) লিনার ভার্যাল মেমোরি সাপোর্ট করে।
- i) লিনারে বিট ইন স্টেট সাপোর্ট সিস্টেম রয়েছে।
- j) ইয় একটি প্রেয়ার্জ লাইনের ও X-Window System এর সাপোর্ট রয়েছে।
- k) লিনার ব্যবহারে বিভিন্ন হার্ডওয়ার এর সর্বোচ্চ পারফরমেন্স পাওয়া যায়।
- l) লিনার GNU-Software কে সাপোর্ট করে।

(গ) ইয় ইনিক্স অপারেটিং সিস্টেম (Unix Opereting System):

১৯৬৯ সালে “Ken Thompson”, “Dennis Rithice” AT & T এর “Bell Labratory” তে DEC PDP-7 নামক মিনি কম্পিউটারের জন্য ইয় ইনিক্সের প্রথম ভার্সন বের করেন। এই ভার্সনটি ছিলো সিস্টেম ইউজারদের জন্য। তবে ইয় ইনিক্সের প্রথম ভার্সন বাজারে আসে ১৯৭১ সালের ৩ নভেম্বর। কিন্তু বৰ্তমানে ইয় ইনিক্স হচ্ছে মাইক্রো, মাইক্রোসোফ্ট, মাইক্রোইনজার, এবং মাইক্রোইনজার অপারেটিং সিস্টেম, যার ফলে বাসিন্ডজিক ও বেজানিক ক্ষেত্ৰে ইয় ইনিক্স বহু ব্যবহৃত হচ্ছে।

ইউনিক্স অপারেটিং সিস্টেমের বৈশিষ্ট্য সমূহ:

- a) ইউজেন্স অপারেটিং সিস্টেমের চিহ্নিতিক (GUI-Graphical User Interface) অপারেটিং সিস্টেম।
- b) ইয় ৩২ বিট ও ৬৪ বিট মাইক্রোপ্রসেসরের সাথে Compatible.
- c) ইয় একটি User Friendly Operating System.
- d) চিহ্নিতিক বা GUI বলে Command গুলো মুছু করতে হয়।
- e) ইহা সাহায্যে খুব সহজে Email, Voice Mail ও Fax Transition এর কাজ করা যায়।
- f) Windows NT, Windows Server Operating System, NOS(Network Operating System) হিসেবে ব্যবহৃত হচ্ছে।
- g) ইয় একটি Single User & Mutitasking Operating System হিসেবে ব্যবহৃত হচ্ছে।
- h) ইয় অপারেটিং সিস্টেম পারসোনাল কম্পিউটার, প্যার্ক স্টেশন, সার্ভার এ ব্যবহৃত হচ্ছে এবং প্রত্যেক ক্ষেত্ৰে গতি বৃদ্ধি ও মেমোরি ম্যানেজমেন্টের উন্নতি সাধন কৰছে।
- i) ইউনিক্স অপারেটিং সিস্টেমের বৈশিষ্ট্য সমূহ:
- j) ইউনিক্স প্রেয়ার্জ কোড কন্ট্রোলে ব্যবহার করা যায়।
- k) ইউনিক্স হায়ারার্কিল ফাইল সিস্টেম ব্যবহার করে। যার ফলে খুব সহজে ও দক্ষভাবে স্বৰূপকৃত রক্ষণাবেক্ষণ ও বাস্তুবায়ন কৰা যায়।
- l) ইউনিক্স পেরিফেরালস ভিডাইসকে ফাইলের মতো মনে কৰে।
- m) একটি সাথে একাধিক ব্যবহারকারী ইউনিক্স অপারেটিং সিস্টেম ব্যবহার কৰতে পারে।
- n) প্রত্যেক ইউজার একটি সাথে একাধিক প্রসেস নির্বাচন কৰতে পারে।
- o) টাইম প্রেয়ারিং কৰে একজন ব্যবহারকারী মাইক্রোইনজিঙ সিস্টেমে কাজ কৰতে পারে।
- p) ইউনিক্স ব্যবহার কৰে গোপনীয়তা রক্ষণ কৰা যায়।
- q) ইউনিক্স যে কোন প্রোগ্রাম ল্যাংগুেজ এর কম্পাইলার ও ইন্টারপ্রেটার সাপোর্ট কৰে।
- r) তেটা বা প্রোগ্রামের সিকিউরিটি খুবই শক্তিশালী।



**প্রশ্ন-১৮. Linux Operating System এর সুবিধা লিখুন?**  
উত্তর: Linux Operating System এর সুবিধা সমূহ:

- a) Low Cost
- b) Stability (স্থায়িত্ব ভালো)
- c) Performance
- d) Network friendliness
- e) Compatibility (সামঞ্জস্যপূর্ণ)
- f) Flexibility (নমুনায়তা)
- g) Fast and Easy Installation
- h) Multitasking
- i) High Security
- j) Open Source OS
- k) Hardware Support
- l) Virtual Memory Support

**প্রশ্ন ১৯: বুটিং (Booting) কি? বুটিং সিকুয়েল (Booting Sequence) তলো কী কী?**

**উত্তর:** বুটিং (Booting): একটি কম্পিউটারকে চালু করা হলে তার CPU বা Central Processing Unit প্রথমেই যাচাই করে নেয়। যে-কি বোর্ড, মাউস, মনিটর ইত্যাদি যোগাশ্ব তার সঙ্গে সঠিক ভাবে যুক্ত আছে কি না। এই যাচাই করার প্রক্রিয়াকে বলে POST, যা Power Of Self-Test নামে পরিচিত।

যদি এই যোগাশ্বগুলি সঠিক ভাবে যুক্ত থাকে, তা হলে CPU কম্পিউটারের অপারেটিং সিস্টেমকে হার্ডডিক থেকে ব্যবহৃত ভাবে রায়মের মধ্যে তুলে নেয় এবং কম্পিউটারকে ব্যবহারকারীর নির্দেশ প্রাপ্তের জন্য প্রস্তুত করে। এই সামগ্রিক প্রক্রিয়াকেই বলা হয় বুটিং (Booting)। অর্থাৎ বুটিং একটি ব্যবহৃত প্রক্রিয়া, যা সম্পূর্ণ হওয়ার পরেই কম্পিউটারকে ব্যবহার করা হয়।

**বুটিং সিকুয়েল (Booting Sequence):** মূলত ডকুমেন্ট রিলেটেড বুটিং তলো সাধারণত বিভিন্ন করা হয়। যদিও তারা কিছু নির্দিষ্ট অপারেটিং সিস্টেমের (যেমন লিনাক্স, উইন্ডোজ মেশিন ইত্যাদি) সাথে সম্পর্কিত। তাই আমরা চোটা করবো সাধারণ সিকুয়েল তলো ফলো করার জন্য। সাধারণত বুটিং সিকুয়েল নিম্নলিখিত ধাপ তলো নিয়ে গঠিত।

- Turn on the Power button.
  - CPU pins are reset and registers are set to specific value.
  - CPU jump to address of BIOS (0xFFFFF0).
  - BIOS run POST (Power-On Self-Test) and other necessary checks.
  - BIOS jumps to MBR (Master Boot Record).
  - Primary Boot loaders run from MBR and jumps to Secondary Boot loader.
  - Secondary Boot loaders load Operating System.
- মূলত বুটিং প্রসেস চলাকালীন সময়ে এই টাস্ক (Tasks) তলোই সম্পূর্ণ হয়ে থাকে।

**প্রশ্ন ২০: Access Time, Seek Time এবং Transfer Time কাকে বলে?**

**উত্তর:** Access Time (একসেস টাইম): কম্পিউটারের তথ্য স্টোরেজ (যেমন- হার্ড ড্রাইভ, এসএসডি, ডিভিডি রম, মেমৰী কার্ড ) থেকে গেট হতে যে সময় লাগে তাকে একসেস টাইম বলে। সাধারণত Access Time ন্যানে সেকেন্ড বা মিলি সেকেন্ডে মাপা হয়।

**Seek Time (সিক টাইম):** একটি Hard Disk Controller একটি নির্দিষ্ট তথ্যের ছান চিহ্নিত করতে যে সময় লাগে, তাকে Seek Time (সিক টাইম) বলে। যখন হার্ডডিকে কোন তথ্য পড়ার অবস্থার দরকার হয় তখন Read/Write হেডকে সঠিক ছান চানসিলিং সিকিং (Seeking) করে। হার্ডডিকে Read/Write হেড-এর সঠিকভাবে অবস্থানে Read/Write হেড-এর ছানসিলিং হওয়ার সময়কে সিক টাইম (Seek Time) বলে।

**Transfer Time: Transfer Time (ট্রান্সফার টাইম):** হচ্ছে ডাটা ট্রান্সফার সম্পর্ক হওয়ার জন্য একটা আনুমানিক টাইম। কারণ অবিকলে ডাটা ট্রান্সফার এর টাইম নির্দিষ্ট না। যখন ডাটা ট্রান্সফার ঘটে তখন ট্রান্সফার টাইম বৃক্ষ পেতে পারে আবার ড্রাইভ ও পেতে পারে।

**প্রশ্ন ২১: What do you mean by firmware?** [এবনী কল্যাণ বৈদেশিক কর্মসূচীন কর্মসূচীন ব্যৱো-২০১৮]

**উত্তর:** কম্পিউটার তৈরি করার সময় তার মেমোরিতে যেসকল প্রয়োজন সমূহ ছান্নাতাবে সংরক্ষণ করে দেওয়া হয়, তাদেরকে ফার্মওয়্যার বলে। বিভিন্ন হার্ডওয়্যার ডিভাইস এর মধ্যে যোগাযোগের জন্য প্রযোজ্য instruction প্রদান করে। চাইলেও এসকল প্রযোজনের কোন পরিস্থিতি করা যান। মূলত এই কাজটা করেন কম্পিউটার তৈরীকারক কোম্পানি বা যারা ম্যাক্যুলারিকার এর দায়িত্ব থাকেন তারা। যেমনং BIOS

**প্রশ্ন ২২: What is boot loader? Why is it use?** [এবনী কল্যাণ ও বৈদেশিক কর্মসূচীন কর্মসূচীন ব্যৱো-২০১৮]

**উত্তর:** যখন কম্পিউটার চালু করা হয় তখন অপারেটিং সিস্টেম লোড করতে একটি প্রযোজন ব্যবহার হয় তাকে বুট লোডার বলে। সাধারণত বুট লোডারের কার্যক্রমকে বুটিং বলা হয়।

একটি কম্পিউটারকে চালু করা হলে তার CPU বা Central Processing Unit প্রথমেই যাচাই করে নেয়। যে-কি বোর্ড, মাউস, মনিটর ইত্যাদি যোগাশ্ব তার সঙ্গে সঠিক ভাবে যুক্ত আছে কি না। এই যাচাই করার প্রক্রিয়াকে বলে POST, যা Power Of Self-Test নামে পরিচিত।

যদি এই যোগাশ্বগুলি সঠিক ভাবে যুক্ত থাকে, তা হলে CPU কম্পিউটারের অপারেটিং সিস্টেমকে হার্ডডিক থেকে ব্যবহৃত ভাবে রায়মের মধ্যে লোড করে নেয় এবং কম্পিউটারকে ব্যবহারকারীর নির্দেশ প্রাপ্তের জন্য প্রস্তুত করে। এই সামগ্রিক প্রক্রিয়াকেই বলা হয় বুটিং (Booting)। অর্থাৎ বুটিং একটি ব্যবহৃত প্রক্রিয়া, যা সম্পূর্ণ হওয়ার পরেই কম্পিউটারকে ব্যবহার করা হয়।

### প্রতিক লিখাত

**প্রশ্ন-১. Linux কি? অথবা Linux কি ধরনে OS?**

**উত্তর:** Linux হলো Multiuser এবং Multitasking Operating System।

**প্রশ্ন-২. Linux Kernel কি?**

**উত্তর:** Linux Kernel: Operating System এর Resident Program হল Kernel। Kernel কে Heart of OS বলা হয়। Kernel হচ্ছে OS এর চালিকা শক্তি। Linux Kernel হলো একটি সফটওয়্যার, যা Application সমূহকে কম্পিউটার হার্ডওয়্যার এর সাথে যোগাযোগ করিয়ে দেয় এবং অন্যান্য Application এর সাথে কাজ করতে সহায়তা করে।

**প্রশ্ন-৩. Linux এর কিছু ডিস্ট্রিবিউশন এর নাম লিখুন?**

**উত্তর:** Linux এর কিছু ডিস্ট্রিবিউশন এর নাম:

- a) Red Hat
- b) Slackward Linux
- c) Linux mandarke
- d) SUSE Linux
- e) Debian Linux
- f) Will Linux etc.

**প্রশ্ন-৪. Linux Installation এর Class তলো লিখুন?**

**উত্তর:** Linux Installation এর ক্লাস তলো নিম্নরূপ:

- a) Personal desktop
- b) Workstation
- c) Server class
- d) Custom class

**প্রশ্ন-৫. Linux এর Desktop Environment তলোর নাম লিখুন? [Jnr. Ins BPSC-2019]**

**উত্তর:** Linux এর Desktop Environment তলো:

- a) GNOME (GNU Network Object Model Environment)
- b) KDE (K-Desktop Environment)

### Types of Operating System Process Management: Process states PCB, Job and Process Scheduling

**প্রশ্ন-১. JCL কি? JCL এর ব্যবহার ও উদ্দেশ্য লিখুন।** [NTRCA(Lecturer)-2012]

**উত্তর:** JCL হচ্ছে Job Control Language এর সংক্ষিপ্ত রূপ। মূলত ব্যাচ প্রসেসিং সিস্টেমে এক প্রযোজন থেকে অন্য প্রযোজন ট্রানজিট (Transit) করার ক্ষেত্রে বা নতুন ডিভাইস ব্যবহারের ক্ষেত্রে অপারেটিং সিস্টেমের জন্য প্রস্তুত করে। এই ধরনে Data ও Code দুটীই থাকে। কিন্তু একই সঙ্গে বিভিন্ন প্রসেস ডেটাকে শেয়ার করলে ডেটা সমূহে অসামঞ্জস্যতা সৃষ্টি হতে পারে। আর যে পক্ষতে এই ধরনের অসামঞ্জস্যতা দূর করা যায় তাকে Process Synchronization Problem (প্রসেস সিনক্রেনাইজেশন সমস্যা) বলে। বিভিন্ন ধরনের Process Synchronization Problem (প্রসেস সিনক্রেনাইজেশন সমস্যা) রয়েছে। যেমন:

- a) The Critical Section Probem (ক্রিটিক্যাল সেকশন সমস্যা)
- b) The Readers-Writers Problem (রিডার-রাইটার সমস্যা)
- c) Dining Philosopher (ডাইনিং পিলোশোপার)
- d) Semaphore (সেমাফের)
- e) Peterson's Solution (পিটারসনের সমাধান)

**JCL বা Job Control Language এর ব্যবহার:** যেকোন প্রযোজনকে কম্পিউটার দ্বারা নির্বাচিত (Execute) করতে হলে তার জন্য

অয়েজনীয় ট্রালেটের (যেমন: Compiler, Assembler, Interpreter ইত্যাদি) এর দরবার হয়। ট্রালেটেরের কাজ হলো সোর্স কোডকে মেশিন কোডে কনভার্ট করা।

আমরা জানি ব্যাচ মোড অপারেটিং সিস্টেমের ক্ষেত্রে অনেক তলো কে Stack (LIFO) আকারে কম্পিউটারে প্রবেশ করানো হয়। আর অন্যেকটি জবকে (Job) পৃথক করার জন্য, আলাদা আলাদা জবের (Job) জন্য সঠিক কম্পাইলার সিলেক্ট করার জন্য Error Detection (ক্রটি নির্দেশ) এবং সে অনুযায়ী দেওয়ার জন্য কিছু Instruction (নির্দেশ) এর দরবার হয়। এই ইন্ট্রোকশন তলো যে ভাষায় দেওয়া হয় তাকেই JCL বা Job Control Language বলে।

**JCL বা Job Control Language** এর কাজ: JCL বা Job Control Language এর কাজ হলো একটি জব (Job) থেকে আলাদা করা এবং জবের (Job) জন্য প্রযোজনীয় উপকরণ সরবরাহ করা। সাধারণত প্রত্যেকটি প্রযোজন এবং ডেটার পূর্বে ওপরে JCL Statement বসানো হয়।

**প্রশ্ন-২. Process (Process)** এর ক্ষেত্রে রেস ক্রিশন (Race Condition) বলতে কী বুঝায়? বা রেস ক্রিশন কী? এর ফলে সৃষ্টি সমস্যা কিভাবে দূর করা যায়? বা প্রসেস এর ক্ষেত্রে রেস ক্রিশন কখন হয়?

**উত্তর:** দুইটি প্রসেস এর পিডিউলিং যখন কোন Critical Condition বা চরম সংকটপূর্ণ হয় যে, পিডিউলিং এবং বিভিন্ন ইন্ট্রোকশন এর কারণে বিভিন্ন রকম ফলাফল প্রদান করে। দুই বা ততোধিক প্রসেস এর প্রত্যক্ষ বা পরোক্ষ (Explicit or Implicit) Data Sharing বা Resouce Sharing এর ফলে রেস ক্রিশনের সৃষ্টি হয়। দুই বা ততোধিক প্রসেস একই সময়ে কোন শেয়ার্ড ডিভাইস (Shared Device) বা ফাইলকে Access করতে চায়, তখন কোন প্রসেসটি আগে Access (অনুমতি) পাবে তা সম্পূর্ণভাবে নির্ভর করে প্রসেস সম্বন্ধের অবস্থার উপর।

**অর্থাৎ** যে প্রসেসটি আগে রেস (Race) করছে সেই প্রসেসটি আগে সার্ভিস পাবে এবং অন্যটি এই সময়টাতে কোন সার্ভিস পাবেনা। মাল্টিপ্রোসেস সিস্টেমের ক্ষেত্রে একেপ প

**প্রশ্ন-৪. Process Synchronization Problem** এর Semaphore (সেমাফোর) বর্ণনা করুন। [কালিপ্রি শিক্ষা অধিদল-২০১৯-জুনিয়র ইলেক্ট্রোকটেকনিক্স ও পাওয়ার]

**উত্তর:** Semaphore (সেমাফোর) এর বর্ণনা: প্রসেস সিনক্রেনেশনের ক্ষেত্রে উন্নতপূর্ণ একটি টুলস হলো Semaphore (সেমাফোর)। একটি Semaphore 'S' হলো এমন একটি পূর্ণ সংখ্যার চলক যা ইন্টিগার টাইপ ভেইলেবল যা কেবলমাত্র দুটি Atomic Operation Wait() ও Signal() এর মাধ্যমে পরিবর্তন করা যায়। এক্ষেত্রে Wait() Operation কে P ঘোষণা এবং Signal() Operation কে V ঘোষণা সূচিত করা হয়। নিচে অপারেশন দুটির Structure বা Syntax দেখানো হলো।

**Wait() এর Structure বা Syntax:**

```
Wait(S)
While(S<=0)
; //Wait is Busy
S.....
```

এবং Signal() এর Structure বা Syntax:

```
Signal(S){
    S++;
}
```

সেমাফোর এর মান কেবলমাত্র Wait() এবং Signal() এর মাধ্যমে পরিবর্তন হবে। যখন একটি প্রসেস কোন সেমাফোর এর মান পরিবর্তন করে তখন অন্যকোন প্রসেস Semaphore এর মান পরিবর্তন করতে পারবেনা।

**প্রশ্ন-৫. Reader-Writer সমস্যা সমাধানে Semaphore কীভাবে ব্যবহৃত হয়? উদাহরণসহ লিখুন।** [কালিপ্রি শিক্ষা অধিদল-২০১৯ পদের নাম: জুনিয়র ইলেক্ট্রোকটেকনিক্স ও পাওয়ার]

**উত্তর:** Reader-Writer সমস্যা সমাধানে Semaphore এর ব্যবহার: যদি কোন ডেটাবেজ একাধিক প্রসেস শেয়ার করে তবে কিছু প্রসেস শুধুমাত্র ডেটা Read করে আর অন্য কিছু প্রসেস ডেটা Read ও Write দুটোই করে। একাধিক প্রসেস একই সাথে ডেটা Read করলে কোন সমস্যা হবেনা। কিন্তু যদি কোন Writer কোনে ডেটাবেজে Write Operation করতে থাকে এবং তখন অন্যকোন Reader বা Writer Process ডেটাবেজকে Access করতে চায় তখন সমস্যার সম্ভাবনা হতে হ্যাঁ এবং এই সমস্যা সমাধানের জন্য Semaphore (সেমাফোর) ব্যবহার করা হয়। প্রথমে সকল Reader Process একটি ডেটা স্ট্রাকচারকে শেয়ার করে:

```
Semaphore rw-mutex=1;
Semaphore mutex =1;
Int read_count=0;
```

তারপরে Semaphore mutex এবং rw-mutex এর মান 1 এবং read\_count এর মান 0 রাখা হয়। rw-mutex টি উভয় Reader & Writer Process ব্যবহার করে।

```
do{
    wait (rw-mutex);
```

/\*Writing is Performed\*/

```
    signal(rw-mutex);
}
while(true);
```

উপরে একটি Writer Process এর কোড দেখানো হলো এবং নিচে Reader Process এর কোড দেখানো হলো। এইক্ষেত্রে উল্লেখ মেইন্টেনেন্স যদি Writer তার Critical Section এ থাকে এবং n Readers অপেক্ষা করতে থাকে তখন একজন Reader rw-writer এর দ্বারা এবং n-1 Readers অপেক্ষা করবে Mutex এ। তারপর যখন একটি Writer তার Signal(rw-mutex) code execute করবে তখন আবার পুনরায় সকল অপেক্ষারত Reader Run করা হবে অবশ্যই Writer কে Signal প্রদান করা যায়। এইক্ষেত্রে বাইবেন কাজটি করে Scheduler।

```
do{
    wait(mutex);
    Read_count++;
    If(read_count==1)
        wait (rw-mutex);
        signal(mutex)
```

/\*Reading is Performed\*/

```
wait(mutex);
read_count--;
if(read_count ==0)
    signal(rw-mutex);
    signal(rmutex);
```

```
}
```

write(true);

**প্রশ্ন-৬. Process Synchronization Problem এ Critical Section Problem বর্ণনা করুন।**

**উত্তর:** মনেকরি, একটি সিস্টেমে n সংখ্যক প্রসেস ( $P_0, P_1, P_2, \dots, P_{n-1}$ ) রয়েছে। প্রত্যেকটি প্রসেসের এমন একটি Code Section রয়েছে যাকে বলা হয় Critical Section এবং যা কোন Common Variable (সাধারণ চলক) কে পরিবর্তন করে। কোন টেবিল অপেক্ষা করে বা কোন ফাইলে Write করে। কিন্তু একাধিক প্রসেস যদি একইসাথে উক্ত সেকশন Run করতে চায় তখন সমস্যার সৃষ্টি হ্যাঁ। তাই দুটি প্রসেসকে কথনে একই সময়ে ঐ অংশ নির্বাচ করতে দেওয়া হবে। প্রত্যেক প্রসেসকে অবশ্যই তার Critical Section Run করার জন্য অনুমতি নিতে হ্যাঁ। যে অংশ অনুমতি নিয়ে কাজ করে তাকে Entry Section কোন হ্যাঁ। তখন অনুমতি পেলে Critical Section এ প্রবেশ করে এবং কাজ শেষ করে Exit Section এর মাধ্যমে বের হ্যাঁ। প্রবর্তীতে অন্য অংশ Run করে। নিচের চিত্রে এর একটি সাধারণ গঠন দেখানো হলো:

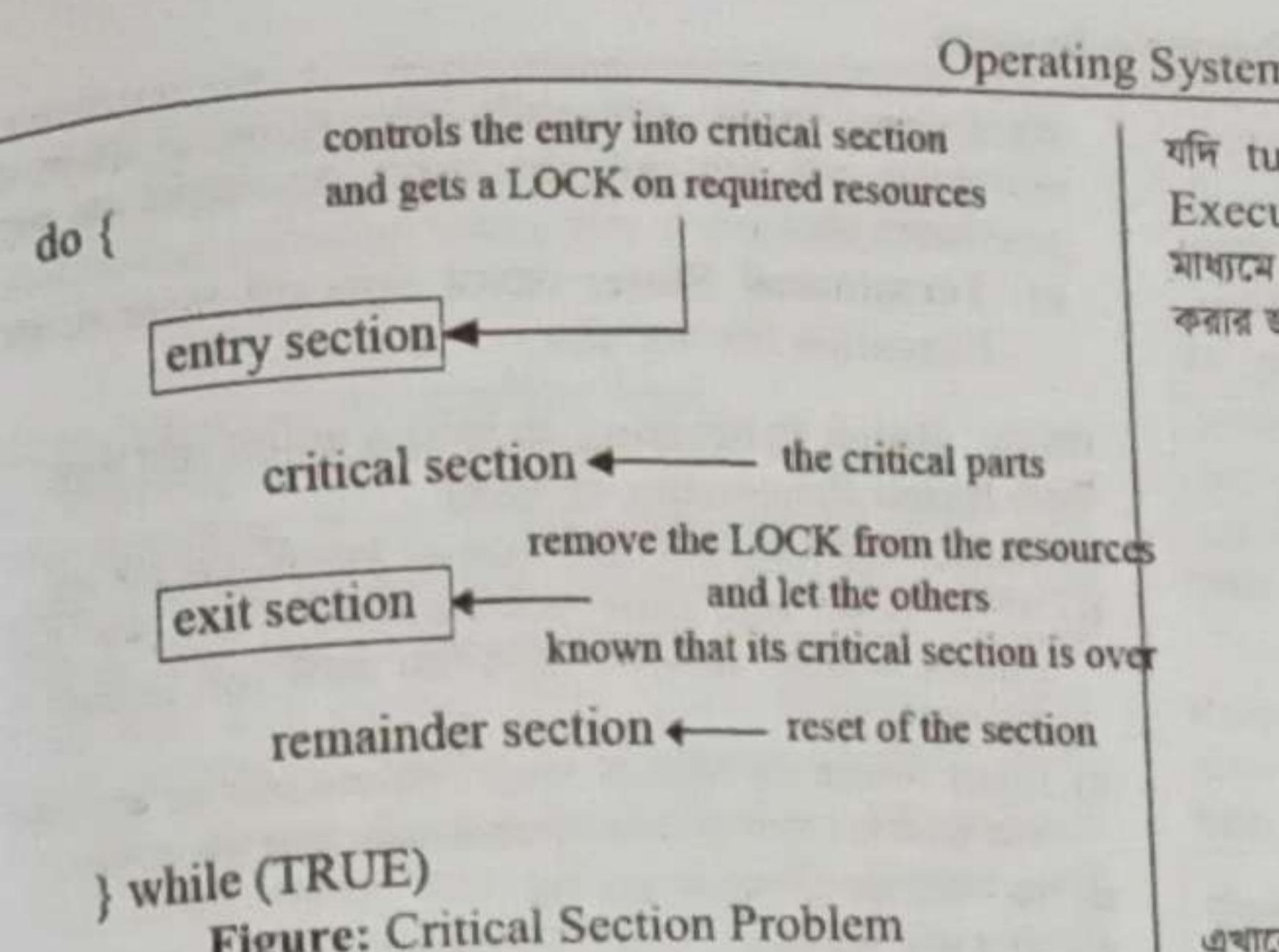


Figure: Critical Section Problem

Critical Section Problem এর যেকোন সমাধানের জন্য ও টি Pre-Condition বা পূর্ণশৰ্ত রয়েছে। যেমন:

(i) **Mutual Exclusion:** যদি কোন প্রসেস  $P_i$  তার Critical Section Run করে তখন অন্য কোন প্রসেস Mutual করে তার Critical Section এ প্রবেশ করতে পারবেনা।

(ii) **Progress:** যদি কোন প্রসেস তার Critical Section এ Run না করে এবং অন্যকিছু প্রসেস Critical Section এ প্রবেশ করতে অঘোষী, তখন যেসব প্রসেস তার Remainder Section এ Run করেনি তারাই কেবল প্রবর্তীতে Run করার জন্য অপেক্ষা করবে।

(iii) **Bounded Waiting:** এ পর্যায়ে একটি Bound বা Limit (সীমা) থাকবে। যার অর্থ অন্য প্রসেস তুলো কর সময় ধরে Wait (অপেক্ষা) করবে, Request করে Critical Section এ প্রবেশ করার অনুমতি পাওয়া পর্যন্ত।

**প্রশ্ন-৭. Process Synchronization Problem এর Peterson's Solution (পিটারসনের সমাধান) বর্ণনা করুন।**

উত্তর: প্রসেস সিনক্রেনেশনের ক্ষেত্রে Critical Section Problem এর সমাধানের মধ্যে একটি সমাধান (Solution) রয়েছে। যা Peterson's Solution (পিটারসনের সমাধান) নামে পরিচিত। কিন্তু আধুনিক কম্পিউটার আর্কিটেকচার মেশিন ল্যাঙ্গুেজ ইলেক্ট্রোকশন নিয়ে কাজ করে বিধায় পিটারসনের সমাধান সঠিকভাবে কাজ নাও করতে পারে। তবে ইহা এমনভাবে ডিজাইন করা হয়েছে যে Mutual Exclusion, Progress এবং Bounded Waiting এই তিনিটি অবস্থাকে নিশ্চিত করে।

সাধারণত পিটারসনের সমাধান 2 টি প্রসেসের মধ্যে সীমাবদ্ধ থাকে এবং তারা একটি অন্যটির বিপরীতক্রমে Execute (নির্বাচ) হ্যাঁ। এক্ষেত্রে প্রসেস দুটিকে  $P_0$  ও  $P_1$  দ্বারা প্রকাশ করা হ্যাঁ। যদি একটি প্রসেস হ্যাঁ  $P_i$  তখন অন্যটি হ্যাঁ  $P_j$ , [শর্ত:  $J=1-i$  হ্যাঁ।] পিটারসনের সমাধানে 2 টি প্রসেস 2 টি ডেটাকে শেয়ার করে। যেমন:

```
int turn;
broken Flag[2];
```

যদি  $turn=i$  হ্যাঁ, তখন প্রসেস  $P_i$  কে তার Critical Section Execute করার অনুমতি প্রদান করা হ্যাঁ এবং Flag Array এর মাধ্যমে নির্দেশ করা হ্যাঁ কোন প্রসেস Critical Section Execute করার জন্য প্রস্তুত রয়েছে। নিচে এর স্ট্রাকচার দেখানো হলো:

```
do{
    flag[i]=true;
    turn=j;
    while (flag[j] && turn==j);
    Critical Section
    flag[i]=false;
    remainder section
}
while(true)
```

এখানে প্রসেস  $P_i$  যখন তার Critical Section এ প্রবেশ করতে চায় তখন প্রথমে তার flag[i] কে true (সত্য) করে এবং turn এর মান j নির্ধারণ করে। কিন্তু যদি দুটি প্রসেসই একজনে তাদের Critical Section Execute করতে চায়, তবে যদিও turn এর মান একজন i এবং অন্যজন j সেট করতে চায় তবুও যেকোন একটি সেট হ্যাঁ এবং তার মান অন্যান্য Critical Section Execute করবে। মোটকোথা turn এর মানই নির্ধারণ করবে যে কোন প্রসেস তার Critical Section Run করার জন্য প্রথমে অনুমতি পাবে।

**প্রশ্ন-৮. Processor Management (Processor Management)** এর ধাপ সমূহ লিখুন।

উত্তর: প্রসেসর হলো একটি হার্ডওয়্যার ডিভাইস, যা Instruction (নির্দেশ) সমূহকে ধাপে সম্পাদন (Execution) করার যোগ্যতা রাখে। সাধারণত প্রসেস এর চাইতে প্রসেসর কর হ্যাঁ থাকে। একটি প্রসেসের একটি প্রসেস বা শেয়ার্ট প্রসেস বা মাল্টিপ্ল প্রসেসকে Execute (নির্বাচ) করতে পারে। দুই বা ততোধিক প্রসেসকে মাল্টিপ্ল প্রসেসের কর্তৃক একজনে নির্বাচ করাকে মাল্টিপ্লেসিং বলে। আবার দুই বা ততোধিক প্রসেসকে একটি প্রসেসের দ্বারা নির্বাচ করাকে মাল্টিপ্রোগ্রামিং বলে।

প্রসেসের ম্যানেজমেন্টের মূল উদ্দেশ্য হলো Over Head Cost সর্বনিম্ন করার জন্য প্রত্যেক প্রসেসকে প্রসেসের সময় Time Slice (সময় ভাগ) এর ভিত্তিতে বরাদ্দ করা। যার ফলে Throughput বৃক্ষি করা যায়। এছাড়াও Subroutine কে শেয়ার করা, Assynchronous (অসামঞ্জস্যপূর্ণ) কার্যকলাপ নিয়ন্ত্রণ করা, বিভিন্ন ধরণের ইন্টারাক্ষন কন্ট্রোল করা প্রসেসের ম্যানেজমেন্টের উদ্দেশ্য।

বিভিন্ন রকমের উদ্দেশ্য ছাড়াও প্রসেসের ম্যানেজমেন্টের কিছু ধাপ আছে। নিচে সেগুলো তুলো ধরা হলো।

- প্রত্যেক প্রসেস এর অবস্থা Store (সংরক্ষণ) করা।
- Ready List থেকে Run করার জন্য প্রসেস Selection করা।
- Inter-Process এর যোগাযোগ Co-Ordinate করা অথবা প্রসেস সমূহের মধ্যে সময়সাময়িক সাধান করা।
- ব্রাউজ্যুক্ত সময় শেষে Running প্রসেসকে সাময়িকভাবে ছাগড়ি রাখা।

প্রশ্ন-৯. Dispatching (ডিস্প্যাচিং) কর্তৃত করুন।  
বা, Dispatcher কাকে বলে?

উত্তর: Dispatcher হলো এমন একটি মডিউল, যা Short term Scheduler দ্বারা নির্ধারিত প্রসেসকে CPU এর নিয়ন্ত্রণ প্রদান করে। Dispatcher এ নির্মাণিত কাজ গুলো থাকে বা Dispatching এর কারণে নির্মাণিত কাজ গুলো হয়।

- a) Switching Context.
- b) Switching to user mode.
- c) Jumping to the proper location in the user program.

প্রশ্ন-১০. Process কাকে বলে? Process State (ধাপ)  
Diagram অংকন পূর্বক করুন। IAM(CTC) NWPGL-2019, বাংলা অনুবাদ- ২০২১-সহকারী প্রয়োজনীয়।

উত্তর: Process: নির্বাচিত কোন প্রোগ্রামকেই প্রসেস বলা হয়। একটি প্রসেস গঠিত হয় মূলত প্রোগ্রাম কাটিন্টারের Current Value (চলমান মান), রেজিস্টার এবং ডেরিয়েবলের সমষ্টিয়ে।

State Of Process: প্রোগ্রাম নির্বাচিত সাথে সাথে প্রসেস তার State পরিবর্তন করে থাকে। অপরদিকে কোন প্রসেসের State দ্বারা তার Current Activity বুকানো হয়। প্রসেস Execution হলো CPU এবং Input/Output Bursts এর একটি Alternating Sequence, যা CPU Burst এর সাথে শুরু ও শেষ হয়ে থাকে। সুতরাং প্রত্যেক প্রসেসই ৫ টি প্রসেসের যেকোনো একটি প্রসেস স্টেটে থাকতে হয়। নিচে চিত্রসহ প্রসেসের ৫ টি State সম্পর্কে আলোচনা করা হলো। যেমন:

- a) New
- b) Running
- c) Waiting
- d) Ready
- e) Terminated

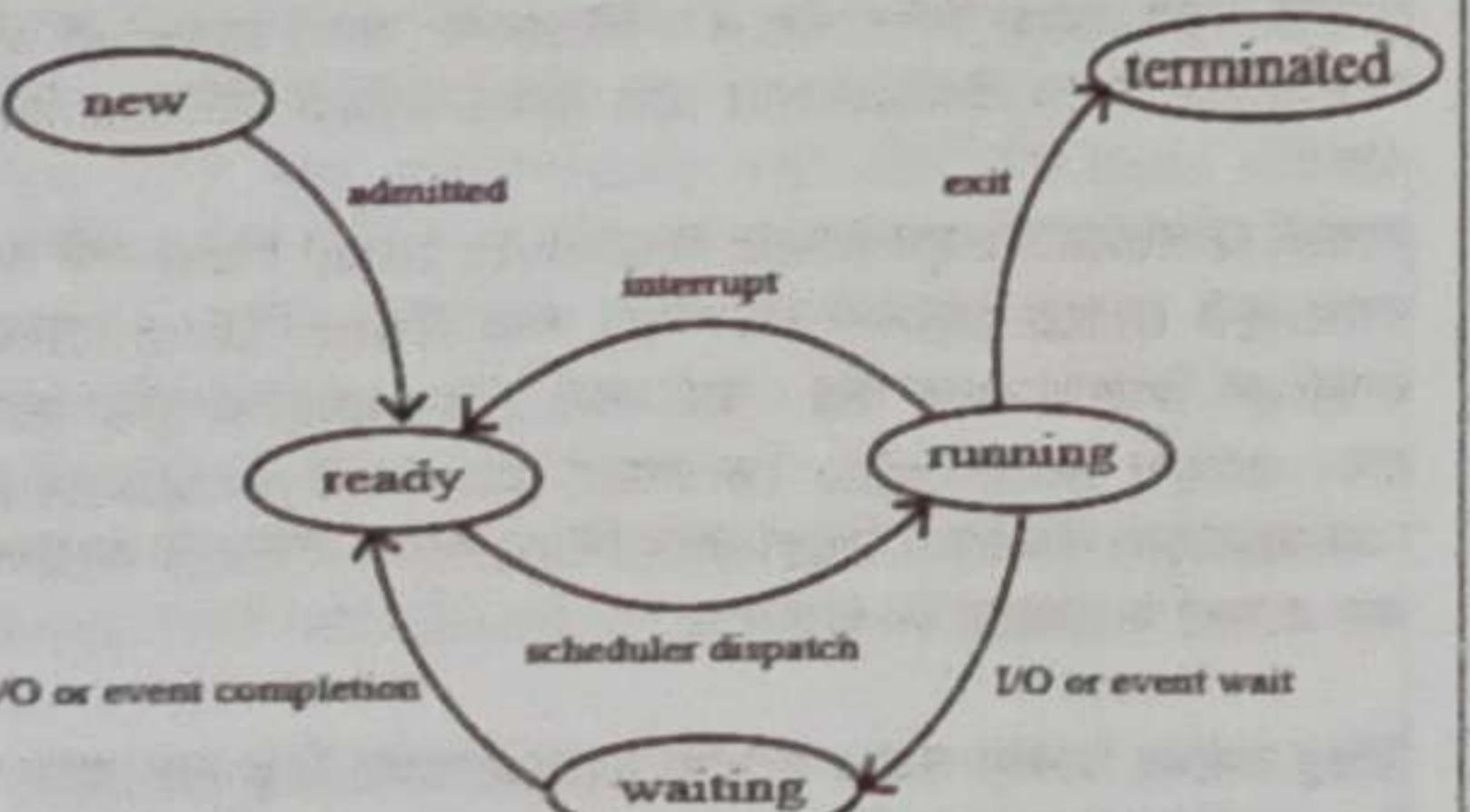


Fig: Diagram of Process state

- a) New State: এই State এ প্রসেসকে Create করা হয়।
- b) Running State: যখন কোন প্রসেস এর Instruction সমূহ প্রসেসের কর্তৃক Sequence অনুসারে সম্পাদন করে।
- c) Waiting State: এই State এ প্রসেস কিছু Event (গটন বা কাজ) সংগঠিত করার জন্য অপেক্ষা করে।

d) Ready State: যখন একটি প্রসেস Running State হাতায়ার জন্য সমস্ত শর্ত পূরণ করেছে এবং প্রসেসের এর অপেক্ষা করাচ্ছে।

e) Terminated State: যেকোন প্রসেস এই State এ প্রক্রিয়া শেষ করে থাকে।

প্রশ্ন-১১. Batch Processing এর সুবিধা ও অসুবিধা তালু লিখুন।

উত্তর: Batch Processing এর সুবিধা:

- a) Batch প্রসেসিং এর ফলে দুইটি Job এর মধ্যবর্তী সময় হ্রাস পায়।
- b) কম্পিউটার এর Idle time কমে যায়। কারণ, যখন কোন Job output এ থাকে, তখন কম্পিউটার অন্য একটি Job Execute এ থাকে।
- c) Input দেওয়ার পর আউটপুট পাওয়া পর্যন্ত অপারেটর এর আর কোন কাজ থাকে না। ফলে সে সময় অপারেটর অন্য কাজ করতে পারে।
- d) এর সময়ে সমস্যা সমাধান করা যায়।
- e) এতে খরচ কম হয়।
- f) Batch প্রসেসিং এর সময় অপারেটর এর মধ্যছাতা করার ধরণ হয় না।

Batch Processing এর অসুবিধা:

- a) ডাটা হারানোর সম্ভবনা থাকে।
- b) অনেকগুলো Job একত্র থাকে বলে ইনপুট দেওয়ার পর নতুন Priority Setup করে দেয়া যায় না। ফলে ডক্টর্চুর্ণ সে Program Complete করতে সময় বেশি লাগে।
- c) Batch মনিটরিং করার জন্য Algorithm এর প্রয়োজন হয়।
- d) ফ্রেম সিস্টেম হওয়াতে তাৎক্ষণিক আউটপুট পাওয়া যায় না।
- e) Priority অনুযায়ী কাজ করা সম্ভব হয় না।
- f) একটি জব শেষ না হওয়া পর্যন্ত আর একটি জব শুরু করা যায় না।

প্রশ্ন-১২. Spooling কীভাবে করুন? Spooling এর পার্শ্ব ব্যাখ্যা করুন।

উত্তর: Spooling: Simultaneous Peripheral Output Line কে সংকেতে Spooling বলে। মূলত Spooling হচ্ছে এক ধরনের প্রসেস (Process)। যার সাহায্যে Batch Mode Operating System এর গতি আরও বৃদ্ধি করা যায়। অর্ধাং বিভিন্ন Peripheral Device এর মধ্যে ডাটা Transmission এর ক্ষেত্রে গতি এবং অসম্ভব্যতা দেখা দেয় তা দ্রু করে CPU এর অলস সময় করা প্রসেসিং এর গতি বৃদ্ধি করার পদ্ধতিকে Spooling বলে।

Process of Spooling: নিচের চিত্রে Spooling এর মাধ্যমে ডাটা ট্রান্সফার হচ্ছে। যখন একটি প্রোগ্রাম কার্ড রিডার থেকে পড়া হয় তখন এটি সরাসরি মেইন মেমোরিতে না গিয়ে ম্যাগনেটিক টেপ বা ডায়ে রয়ে আছে। এরপর ডায়ে থেকে প্রধান স্থৃতিতে নিয়ে নির্বাচিত করা যায়। এরপর প্রাপ্ত ফ্লাক্স সরাসরি আউটপুটে নেয়ার পরিবর্তে ডিক হয়ে আটপুট প্রদান করা হয়।

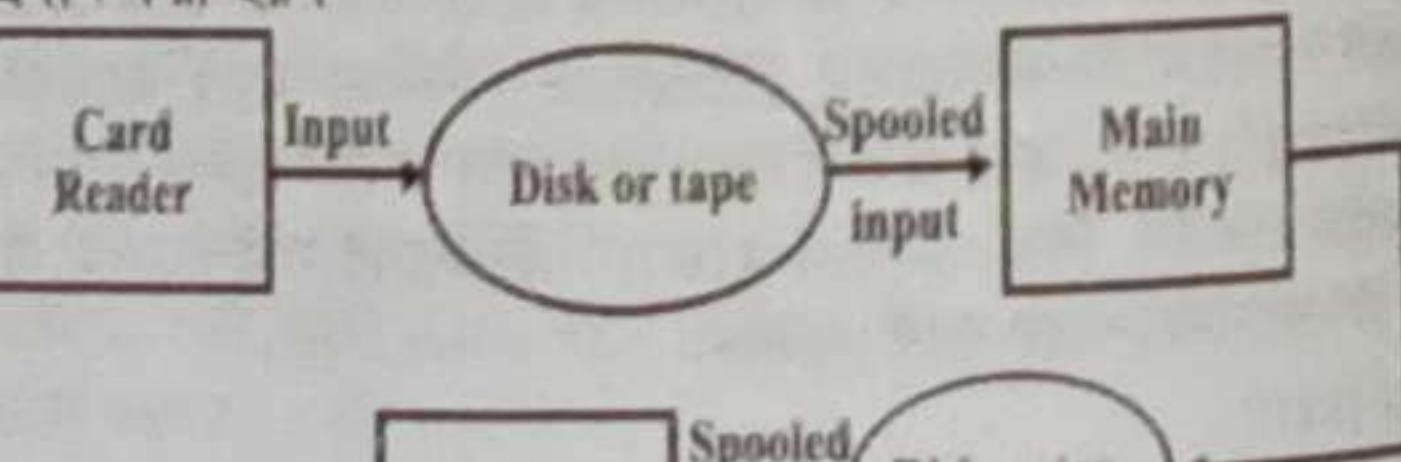


Figure: Spooling পদ্ধতিতে ডাটা ট্রান্সফার।

এখনে লক্ষনীয় যে, কার্ড রিডার অপেক্ষা ডিক বা ট্যাপের গতি অনেক বেশি এবং স্লিপ্টার অপেক্ষা ট্যাপ বা ডিকের গতি অনেক নেশি। অর্ধাং গতি বৃদ্ধি করার জন্য মিডিয়া হিসেবে ট্যাপ বা ডিক ব্যবহার করা হচ্ছে, এবং এটাই এর মূল উদ্দেশ্য।

প্রোগ্রামের মাধ্যমে দুটি প্রসেস কে Swithching বা Associate করা যায়।

একটি প্রোগ্রাম নির্বাচিত করা শেষ না হওয়া পর্যন্ত অন্য প্রোগ্রাম সম্পর্ক হয় না।

প্রসেস একটি Active Entity।

প্রোগ্রাম একটি Passive Entity।

প্রসেস কতগুলো স্টেট (State)

প্রোগ্রাম শুধুমাত্র ব্যবহারকারীর মেনে চলে।

প্রসেস একটি চলমান কার্য।

প্রোগ্রাম চলমান কার্য নয়।

প্রশ্ন-১৩: Spooling এর প্রয়োজনীয়তা লিখুন।  
বা, Spooling কেন দরকার, লিখুন?

উত্তর: Spooling এর প্রয়োজনীয়তা: Spooling এর উদ্দেশ্য হলো প্রোগ্রাম এবং ডাটা সমূহকে দ্রুত গতিসম্পন্ন করা এবং CPU এর ব্যবহারের জন্য Input/Output মাধ্যমকে প্রত্যন্ত রাখা। সাধারণত Dedicated I/O Device যেমন: Card Reader, Printer ইত্যাদি এর কাজ I/O speed, CPU speed এর তুলনায় অনেক কম। দ্রুত গতিসম্পন্ন Card Reader মিনিটে ১৫০০ ক্লামের ১৫০০ Card Read করতে পারে। বলা যায়, প্রতি সেকেন্ডে মোট  $(1500 \times 60) = 20000$  বাইট পড়তে (Read) পারে। তাহলে এর গতি হলো ২০০০ বাইট/সেকেন্ড। অপরদিকে ট্যাপ বা ডিকের গতি হলো ১০০০০০ বাইট/সেকেন্ড। মূলত এই সব Speed Mismatch Remove (দূর) করে। যার ফলে সিপিইউ (CPU) এর আইডল (Idle) টাইম কমে যায় এবং কম্পিউটারের পারফরমেন্স বৃদ্ধি পায়। আর এই জন্যই Spooling এর প্রয়োজনীয়তা অপরিসীম।

প্রশ্ন-১৪: Process এবং Thread মধ্যে পার্শ্বক্য কি? [বিভিন্ন মন্তব্য-২০১৯-সহকারী প্রয়োজনীয় (নম-ক্ষাত্তার)]

উত্তর: Process এবং Thread মধ্যে পার্শ্বক্য

Process	Thread
যে কোন প্রোগ্রাম execution করাকে প্রসেস বলে।	দ্রুত হলো প্রসেসের একটি বিভাগ।
প্রসেস শেষ হতে অনেক সময় লেব।	দ্রুত সমাপ্ত হতে কম সময় লেব।
প্রসেস তৈরির জন্য অনেক সময় লেব।	দ্রুত তৈরির জন্য কম সময় লেব।
কম্পটেক্ট স্যাইচিয়ের জন্য এটি অনেক সময় লেব।	কম্পটেক্ট স্যাইচিয়ের জন্য এটি কম সময় লেব।
প্রসেস অনেক রিসোর্সেস গ্রহণ করে।	দ্রুত অনেক কম রিসোর্সেস গ্রহণ করে।
প্রসেস আইসোলেটেড হয়ে থাকে।	দ্রুত মেরি শেয়ার করে।

প্রশ্ন-১৫. PCB (Process Control Block) Diagram  
অংকন করুন।

উত্তর: Process Control Block (PCB) হল একটি ডাটা স্ট্রাকচার যাতে বিভিন্ন তথ্যাদি জমা থাকে এবং ইহা Process Control Block নামে অপারেটিং সিস্টেম কর্তৃক তৈরি। PCB তে নির্মাণিত তথ্য থাকে।

Pointer	Process State
Process Number	
Process Counter	
Registers	
Memory Limits	
List of Open Files	
:	
:	

Figure: Process Control Block (PCB)

প্রশ্ন-১৬. I/O Bound Program & CPU/Processor Bound Program কাকে বলে?

উত্তর: I/O Bound Program: Multiprogramming System এ যে সকল প্রোগ্রাম প্রসেসিং এর ক্ষেত্রে বেশির ভাগ সময় I/O এর কাজে নিয়োজিত থাকে তাদেরকে I/O Bound Program বলে।

**CPU/Processor Bound Program:**  
Multiprogramming এর ক্ষেত্রে যে প্রোগ্রাম সমূহ প্রসেসিং এর ক্ষেত্রে বেশির ভাগ সময় প্রসেসিং এর কাজে নিয়োজিত থাকে তাদেরকে CPU/ Processor Bound Program বলে।

প্রশ্ন-১৯. **Process Scheduling** কি? এর অকার্ডেন বর্ণনা করুন?  
উত্তর: Process Scheduling এর উদ্দেশ্য হচ্ছে এক বা একাধিক প্রসেসের অনেক সময় সম্পর্ক টাইম। Throughput এবং Processor Efficiency বিচেতন করে যে প্রসেস সমূহ নির্বাহ হবে তাদেরকে একাসাইন (Assign) করা। এই প্রক্রিয়া তিনটি পৃথক ফাংশনে সম্পর্ক হচ্ছে।

**Process Scheduling** এর অকার্ডেন বর্ণনা:  
(i). **Long Term Scheduling:** Long term Scheduling এর অপর নাম Job Scheduling। Long Term Scheduling এর ক্ষেত্রে সিডিলার (Scheduler) নির্ধারণ করে কোন Process টি Processing এর Admitted হবে। Long Term Scheduling এর ক্ষেত্রে Secondary Memory থেকে Main Memory তে Execution Process load হবে।

(ii). **Mid Term Scheduling:** এটি Swapping ফাংশনের অংশ। কোন সিক্ষার্থী ফলাফল হিসাবে এটি যে সমস্ত প্রসেসের অংশ বিশেষ প্রধান স্থানে আছে এবং যে গুলি নির্বাহের অপেক্ষা আছে তাদের সাথে কোন প্রসেসকে যুক্ত করে।

(iii). **Short Term Scheduling:** একে CPU Scheduler বলা হয়। Short Term Scheduling এর ক্ষেত্রে Scheduler Ready Queue থেকে পরবর্তী Process কে Execution এর জন্য নির্ধারণ করে।

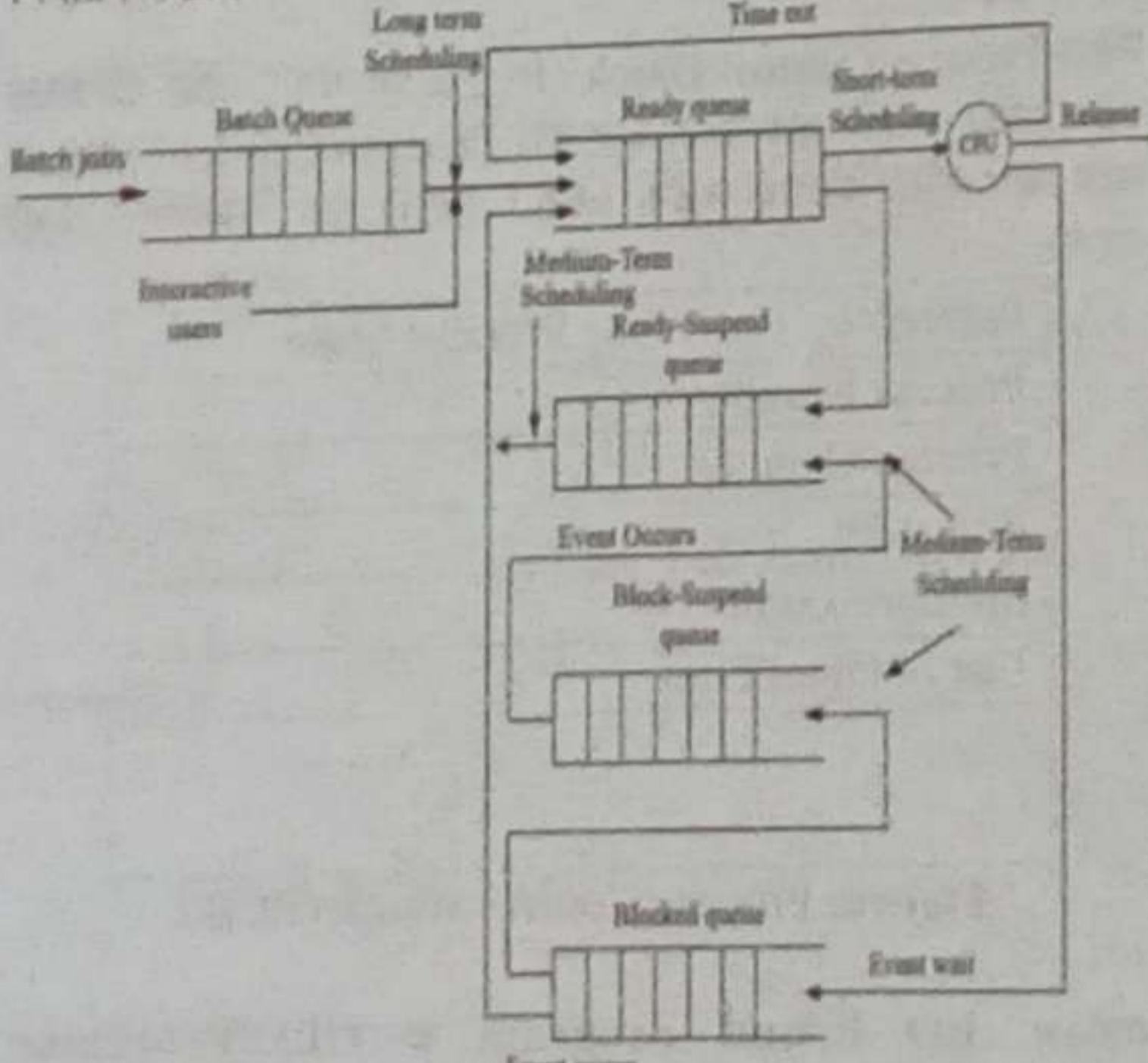


Figure: Queuing diagram for Scheduling

প্রশ্ন-২০. **CPU Scheduling** কাকে বলে? CPU Scheduling এর Criteria কোন বর্ণনা করুন? [NTRCA(Lecturer)-2017]

উত্তর: **CPU Scheduling:** যদি একাধিক Job/Task একই সময় Ready থাকে Run করার জন্য তখন এই সমস্ত Job/Task হতে যে

Job/Task টি Run করবে, তার জন্য যে Decision করা হয় তাকে CPU Scheduling বলে।

**CPU Scheduling এর Criteria সমূহ:**

**CPU Utilization:** CPU Utilization ক্ষেত্রে সুবাহ হচ্ছে। CPU ব্যবহৃত হচ্ছে। ইহার মান ০ হতে 100%।

**Throughput:** এটি একক ইউনিট (Unit) এর মধ্যে যে প্রসেসের অনেক সম্পর্ক হচ্ছে। Throughput এবং Processor Efficiency বিচেতন করে যে প্রসেস সমূহ নির্বাহ হবে তাদেরকে একাসাইন (Assign) করা। এই প্রক্রিয়া তিনটি পৃথক ফাংশনে সম্পর্ক হচ্ছে।

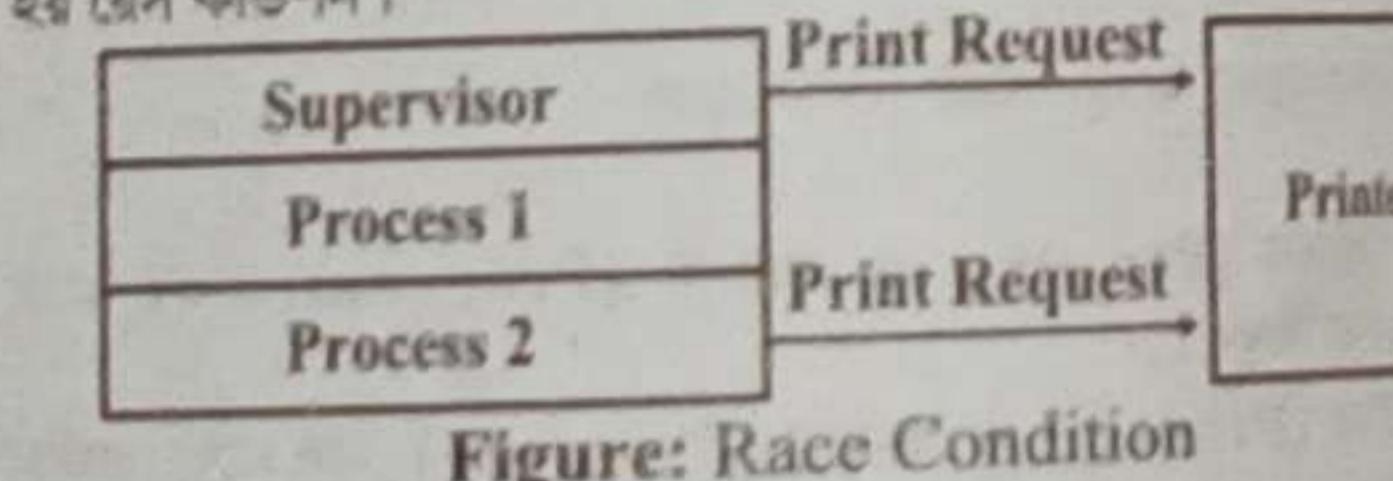
**Turnaround Time:** কোন Job Submission এর সময় নির্বাহ থেকে হওয়া পর্যন্ত মোট সময়কে বলা হয় Turnaround Time। একটি ভালো এলগরিদমের Turnaround Time কর্তৃত ভাল।

**Waiting Time:** যতক্ষণ যাবত কোন Process Ready Queue তে অপেক্ষা করে তাকে Waiting Time বলা হয়। Waiting Time যত হবে, একটি এলগরিদম তত ভালো হবে।

**Response Time:** যে সময় কোন প্রসেস Response করে যে করে। অর্থাৎ Response Time হলো Responding ক্ষেত্রে প্রযোজনীয় সময়। অভ্যন্তরে User এর Response Time যত হবে, তত ভালো।

প্রশ্ন-২১. **Race Condition** কি? চিত্রসহ আলোচনা করুন। [BCS]

উত্তর: Race Condition: যদি দুই বা ততোধিক প্রসেস যারা সময়ে কোন শেয়ার ডিভাইস বা ফাইলকে আয়োজন করতে চান এবং একে অবহৃত কোন প্রসেসটি আগে একের পাবে তা নির্ভর করে যে সময়ের অবহৃত উপর অর্থাৎ যে প্রসেসটি আগে রেস করে যেটি যার সার্ভিস পাবে এবং অন্যটি ঐ সময়ে কোন সার্ভিস পাবে এবং মাল্টিপ্রোসেসিং সিস্টেমের একে অনেক প্রসেস নির্বাচন করার পরিস্থিতিতে হয় রেস ক্রিসেন্স।



প্রশ্ন-২২. **Batch Processing System** (ব্যাচ প্রসেসিং সিস্টেম) কী বুঝেন? চিত্রসহ ব্যাখ্যা করুন।

উত্তর: **Batch Processing System** (ব্যাচ প্রসেসিং সিস্টেম) একটি প্রোগ্রামের কিছু অংশ নির্বাহ (Execute) হওয়ার পর আবার অন্য প্রোগ্রাম এর কিছু অংশ নির্বাহ (Execute) হয়। এভাবে সব প্রোগ্রাম শেষে আবার প্রথম প্রোগ্রাম নির্বাহ (Execute) হয়, এজন প্রোগ্রামের Status সংরক্ষণ করার প্রয়োজন হয়।

ব্যাচ প্রসেসিং সিস্টেমের চিত্রসহ বর্ণনা: এই পদ্ধতিতে একটির পর একটি Application Program (ব্যাবহারিক প্রোগ্রাম) নির্বাহ করা হয়। প্রোগ্রাম নির্বাহের (Execute) সময় কম্পিউটারের সমর্থন (Ability) ও সুযোগ সুবিধা অনুযায়ী তখ্ন এই প্রোগ্রামটির জন্য নির্যোজিত থাকে। তাই এই পদ্ধতিতে দীর্ঘ সময় ধরে ডেটা সংগ্রহ করে পরে একসাথে প্রযোজনীয় প্রসেসিং এর কাজ সম্পর্ক করে। অটোমেটিক্যালি এক Job (জব) থেকে অন্য Job (জব) এ যাওয়ার উপর ভিত্তি করে এই অপারেটিং সিস্টেম তৈরি হয়ে থাকে।

প্রশ্ন-২৪. মাল্টি প্রোগ্রামিং অপারেটিং সিস্টেম কী? সচিত্র বর্ণনা করুন। [NTRCA(Lecturer)-2019]

উত্তর: **Multiprogramming:** যে সব অপারেটিং সিস্টেম থার একটি কম্পিউটারে একসাথে অনেকগুলো প্রোগ্রাম / জব / প্রসেস পরিচালনার কাজ করা যায়। তাকে মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেম বলে।

**সচিত্র বর্ণনা:** কম্পিউটারের Input/Output প্রেরিফেরিয়াল ডিভাইস সমূহ যখন তথ্য আদান-প্রদানে ব্যুৎ থাকে। সিপিইউ (CPU) তখন প্রসেসেইন বা কাজবীল অলস সময় কাটায়। মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেমে CPU এর অলস সময়টাকে কাজে লাগিয়ে একইসাথে একাধিক প্রোগ্রাম পরিচালনার সুযোগ সৃষ্টি করে। ফলে প্রসেসিং এর গতি বেড়ে যায়।

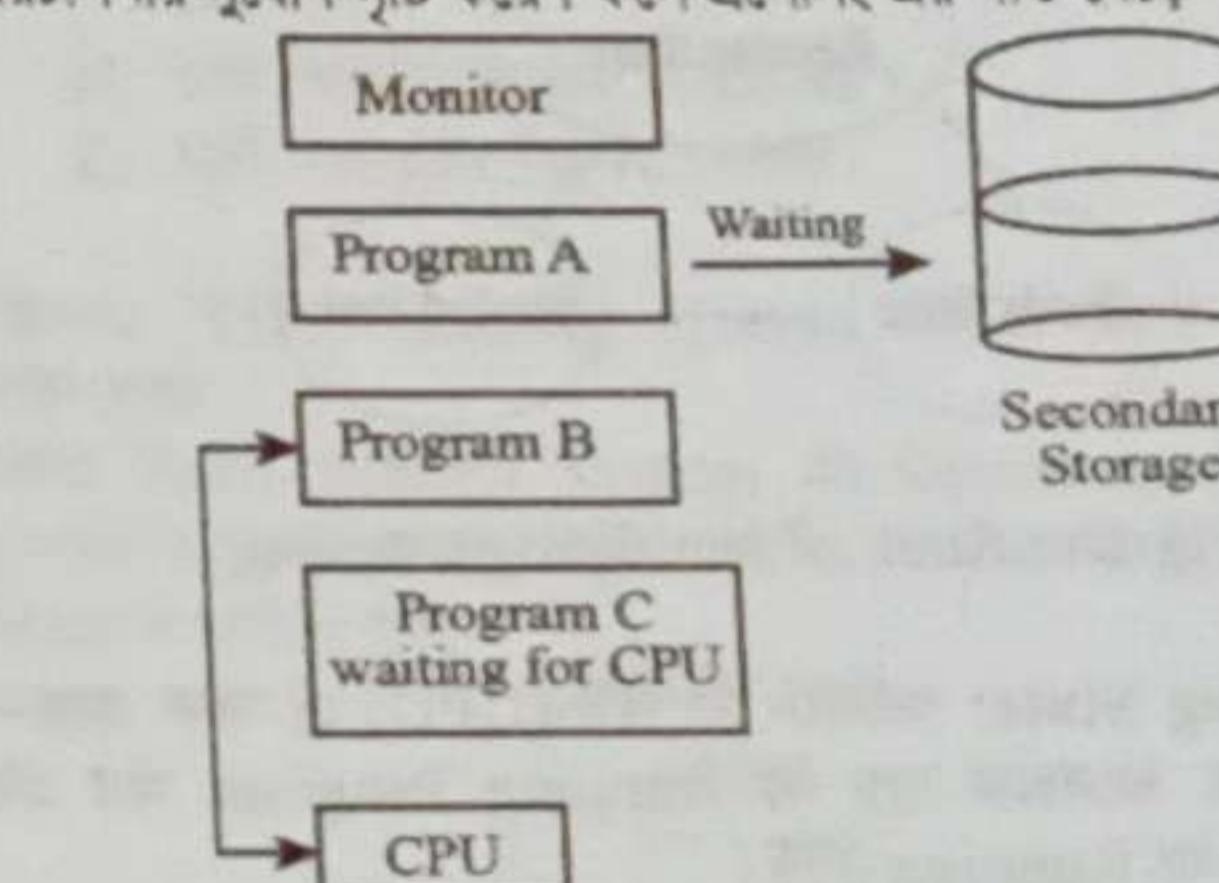


Figure: Multiprogramming Operating system

চিত্রে মাল্টিপ্রোগ্রামিং অপারেশন দেখানো হয়েছে। যেখানে Program-A CPU ব্যবহার করেছে। অর্থাৎ Data O/P এর কাজে নির্যোজিত। Program-B CPU ব্যবহার করে প্রসেসিং কাজে ব্যুৎ এবং Program-C CPU এর জন্য Waiting এ আছে।

প্রশ্ন-২৫. **Multiprogramming এর Requirement** তালো কি কি? বর্ণনা করুন।

উত্তর: **Multiprogramming এর Requirement** তালোর বর্ণনা:  
**Large Memory:** যেহেতু এতে একই সাথে একাধিক ব্যাবহারকারী কাজ করতে পারে তাই ব্রহ্ম মেমোরি এর দরকার হয়।

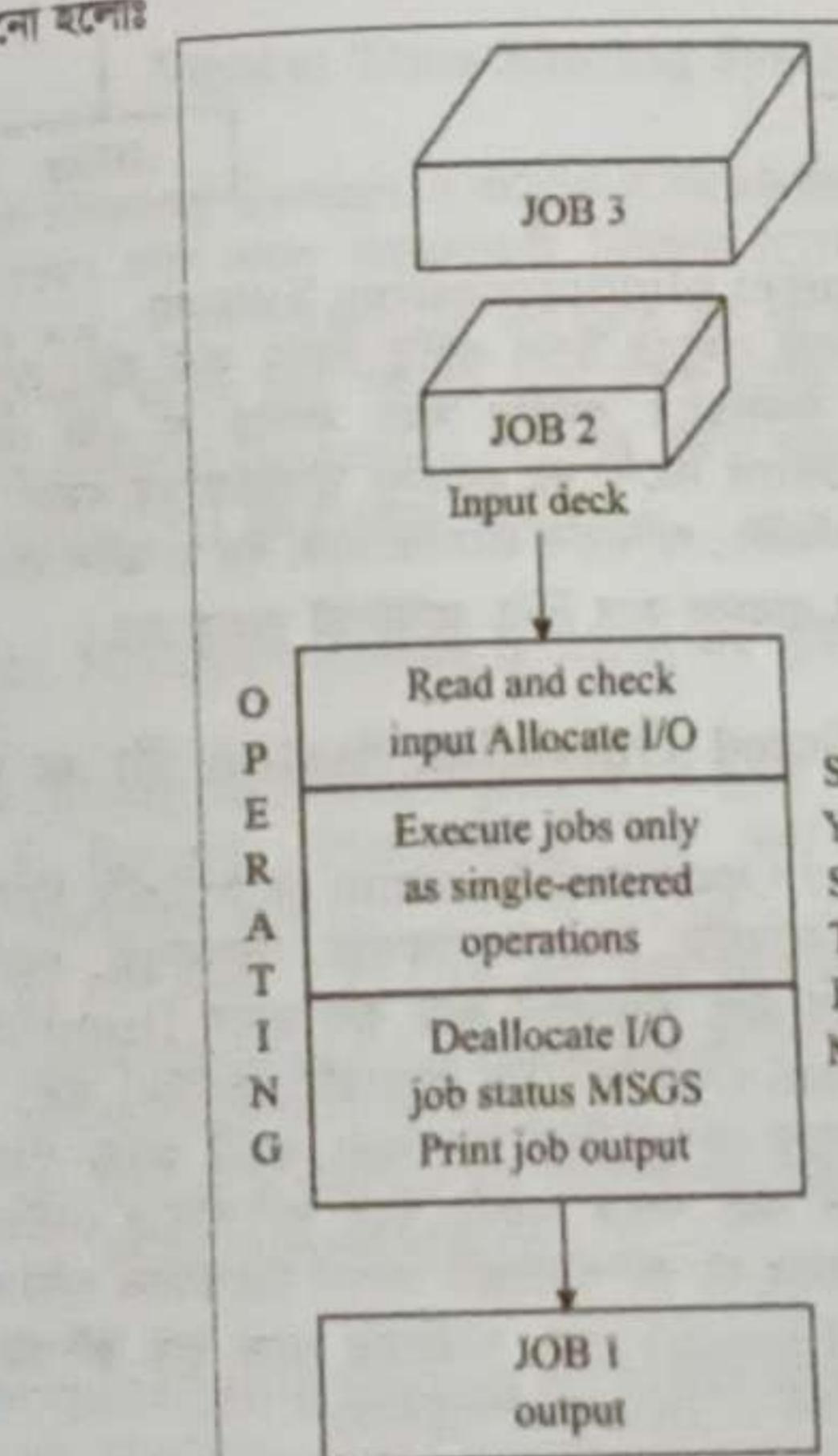


Figure: Serial Batch System Operation

প্রশ্ন-২৬. **Batch Processing** এর রিসোর্স কয়টি ও কি কি? উত্তর: Batch প্রসেসিং এর রিসোর্স দুটি। যথা:

- a) Main Memory
- b) CPU

**Memory Protection:** [36<sup>th</sup> BCS, ICT Ministry - 12]  
Memory Protection একটি হার্ডওয়ার ও সফটওয়ার মেথড যেখানে মেমরিতে ইনফরমেশন রিভ এবং রাইট যেন আনঅ্যাক্সেস ক্রেডিট কেউ আয়োজন করতে না পারে। আবার একটি মেমোরি পার্টিশন এর প্রয়োগ অন্য একটি মেমোরি পার্টিশন এর প্রয়োগ এর উপর যাতে প্রত্যাব ফেলতে না পারে সেজন্য �Memory Protection দরকার।

**Program Status Preservation:** যেহেতু Multiprogramming System এ একটি প্রোগ্রামের কিছু অংশ নির্বাহ (Execute) হওয়ার পর আবার অন্য প্রোগ্রাম এর কিছু অংশ নির্বাহ (Execute) হয়। এভাবে সব প্রোগ্রাম শেষে আবার প্রথম প্রোগ্রাম অন্য একটি মেমোরি পার্টিশন এর প্রয়োগ এর উপর যাতে প্রত্যাব ফেলতে না পারে। এজন প্রোগ্রামের Status সংরক্ষণ করার প্রয়োজন হয়।

**Proper Job Mixing:** CPU নির্ভুল কাজ এবং I/O নির্ভুল কাজ সমূহ এমনভাবে নির্ধারণ করা হয় যাতে সিপিইউ (CPU) সব সময় ব্যবহার করা হোক।

**প্রশ্ন-২৬. Multiprogramming System এর স্টেট কয়টি ও কি কি? চিত্রসহ বর্ণন।**

**উত্তর:** Multiprogramming System এ তিনটি স্টেট এ কাজ সম্পন্ন হয়ে থাকে। যথা:

- a) Running State
- b) Ready State
- c) Blocked State

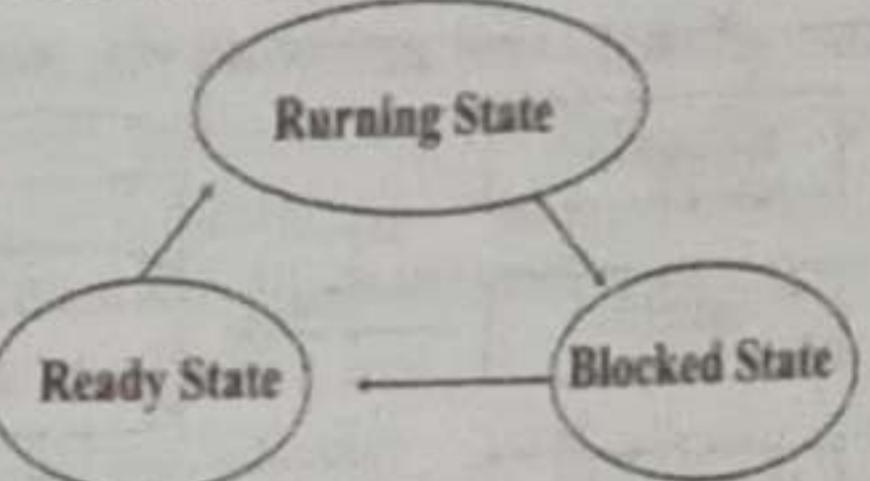


Figure: State of Multiprogramming

a) **Running State:** বর্তমানে সিপিইউ (CPU) যে সকল প্রোগ্রাম ব্যবহার করছে তাদেরকে বলা হয় Running Program আর এই স্টেট কে বলা হয় Running State।

b) **Ready State:** যে সকল প্রোগ্রাম সিপিইউ কর্তৃক নির্বাচিত (Execute) এর জন্য অপেক্ষা করছে তাদেরকে বলা হয় Ready Program আর এই স্টেট কে বলা হয় Ready State।

c) **Blocked State:** যে সকল প্রোগ্রাম এই মাঝে সিপিইউ (CPU) এর ব্যবহার শেষ করেছে বা I/O Operation এর জন্য সিপিইউ (CPU) এর ব্যবহার থেকে বিরত রয়েছে, তাদেরকে Blocked Program বলা হয়। আর এই স্টেট কে বলা হয় Blocked State।

**প্রশ্ন-২৭. Multi Programming এর সুবিধা ও অসুবিধা লিখ।**

**উত্তর:** Multi Programming এর সুবিধা সমূহ:

- a) Throughput বৃদ্ধি পায়।
- b) Response time কমায়।
- c) CPU Idle থাকে না।
- d) মেমোরি খালি থাকে না প্রায় সম্পূর্ণ ব্যবহৃত হয়।
- e) মোট প্রসেসিং সময় অনেক কম স্পীডের কলে অধিনেতৃত দিক থেকে অনেক লাভ জনক হয়।

**Multi Programming এর অসুবিধা সমূহ:**

- a) Operation জটিল।
- b) অতিরিক্ত হার্ডওয়্যার ও সফটওয়্যার প্রয়োজন হয়।
- c) অধিক মেইন মেমোরি এর প্রয়োজন হয়।
- d) তাৎক্ষণিক ফলাফল পাওয়া যায় না।

**প্রশ্ন-২৮. Multiprocessing System কাকে বলে? চিত্রসহ বর্ণনা করন। [36<sup>th</sup> BCS]**

**উত্তর:** Multiprocessing System: যে কম্পিউটার সিস্টেমে একাধিক CPU & Processor ব্যবহার করা হয় তাকে Multiprocessing System বলে।

**চিত্রসহ বর্ণনা:** একটি নির্দিষ্ট সময়ে একটি কাজে একাধিক অস্পৃষ্ট ইন্টেলিজেন্সি সিস্টেমে কাজ করার Ability কে মাইক্রোসিস্টেম বলে। এই System এ একাধিক Processing Unit থাকে যাতে অনেক ক্ষেত্রে Job করা সম্ভব। একেতে একসাথে একাধিক প্রোগ্রাম মেমোরিতে রাখ থাকে। একই প্রোগ্রাম একাধিক প্রসেসরে কিংবা ভিন্ন ভিন্ন অস্পৃষ্ট কম্পিউটারে একাধিক অস্পৃষ্ট ব্যবহার করে একাধিক প্রোগ্রাম নিয়ে কাজ করা যায়।

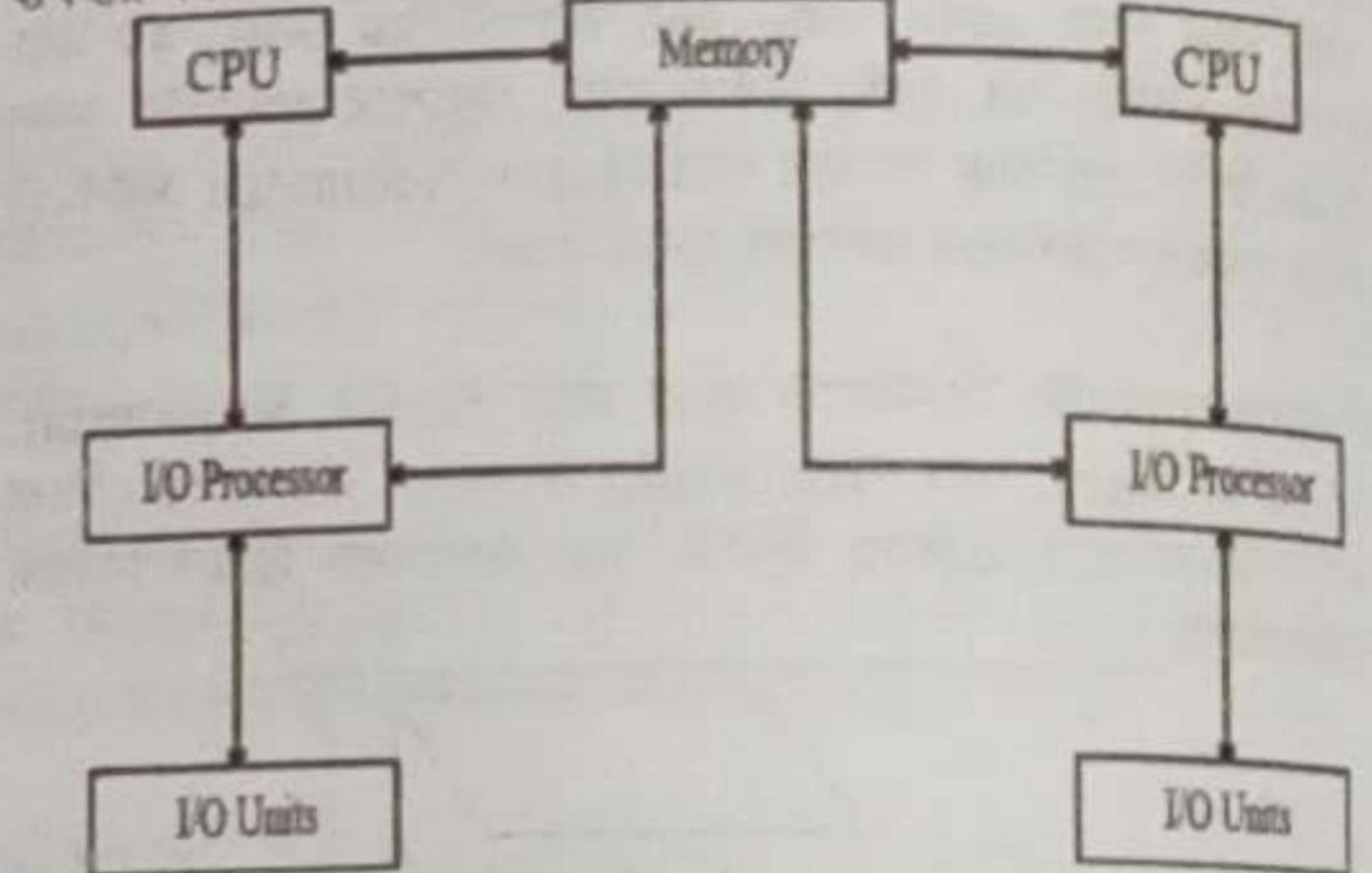


Figure: Multiprocessing System

এই সিস্টেমে প্রসেসর সমূহের উপর উক্ত অস্পৃষ্ট এন্দান করা হয়। একেতে কোনো প্রসেসরই অস্পৃষ্ট কর্মসূচী সময় কাটায় না এবং কোনো প্রসেসর নষ্ট হয়ে গেলেও সিস্টেমের প্রসেসিং কার্যক্রম ব্যবহৃত হয়। এই প্রসেসিং প্রক্রিয়ার এভিটিং, গান্ধিতিক প্রসেসিং এবং বড় ও জটিল প্রসেসিং এর জন্য এই পদ্ধতি ব্যবহার করে দ্রুত আউটপুট পাওয়া যায়।

**প্রশ্ন-২৯. Distributed Operating System কী? এর কাজ তলো লিখুন।**

**উত্তর:** Distributed Operating System: যে অপারেটিং সিস্টেম কমিউনিকেশন নেটওর্কের মাধ্যমে পরিম্পরাগ সম্পর্কসূচী একাধিক কম্পিউটারকে নিয়ন্ত্রণ এবং তত্ত্বাবধান করা যায় তাকে Distributed Operating System (ডিস্ট্রিবিউটেড অপারেটিং সিস্টেম) বলে। এই সিস্টেমের উদ্দেশ্য হলো সম্পূর্ণ সিস্টেমের জন্য একটি অঙ্গ পরিম্পরাগ তৈরি করা এবং যুব অঙ্গ ধরণে একটি বৃহৎ কম্পিউটার নেটওর্ক তত্ত্বাবধান করা। একেতে যে কম্পিউটারটি সম্পূর্ণ সিস্টেমকে পরিচালন করে তাকে সার্ভার (Server) এবং সার্ভারের সাথে যুক্ত কম্পিউটার তলোকে বলা হয় ক্লায়েন্ট (Client) কম্পিউটার।

**Distributed Operating System নির্মাণ কর্তৃপক্ষ করে থাকে। যেমন:**

- a) মেমোরি তত্ত্বাবধান করা।
- b) ইনপুট-আউটপুট তত্ত্বাবধান করা।
- c) ফাইল তত্ত্বাবধান করা।
- d) বিভিন্ন ডিভাইস থেকে CPU তে ডেটা ও ইনফরমেশন অদান প্রদান করা।
- e) ইন্ট্রাকশন প্রসেসিং কে তত্ত্বাবধান করা।

**প্রশ্ন-৩০. Time Sharing System কাকে বলে? চিত্রসহ বর্ণনা করন। [NTRCA (Lecturer)-2013 & 2014]**

**উত্তর:** Time Sharing System: যখন কোন সিস্টেম একের অধিক ব্যবহারকারী দ্বারা ব্যবহৃত হয় তখন ব্যবহারকারীকে সময়ের সাপেক্ষে সিপিইউ (CPU) ভাগ করে দেওয়া হয়। এই সময় ভাগ করে দেওয়াকে Time Sharing System বলে।

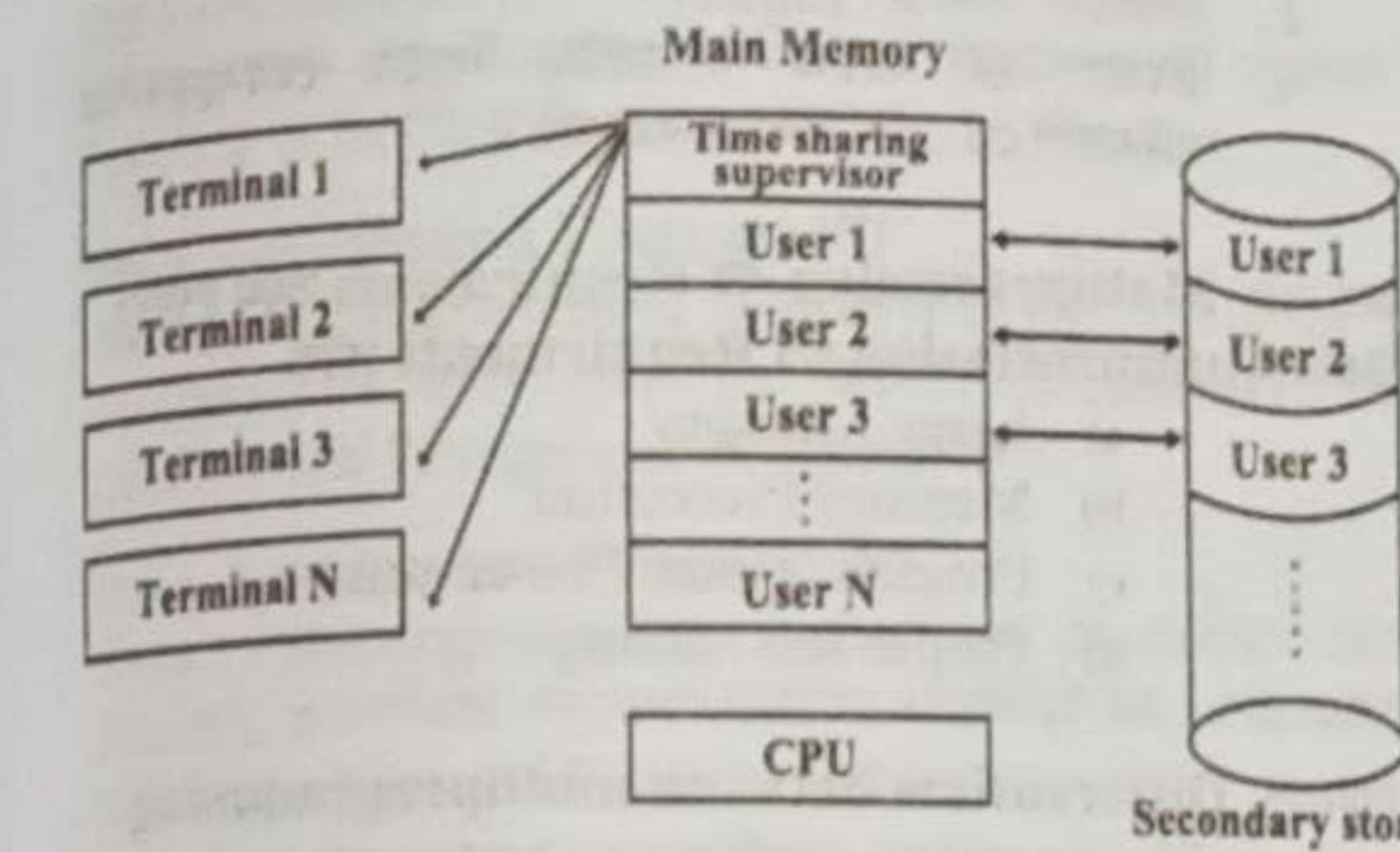


Figure: Time Sharing System

Time Sharing System এ প্রত্যেক ব্যবহারকারীর প্রোগ্রাম কে কিছু সময় দেয়। যাতে সকল ব্যবহারকারী সিপিইউ (CPU) ব্যবহারের সুযোগ পায়। CPU Service এত দ্রুত হয় যে কোন ব্যবহারকারী ব্যবহৃত পারে না যে কিছু সময়ের জন্য তাকে সার্ভিস প্রদান কর হবে। Time Sharing System এ যে কোন একটি মুহূর্তে যে কোন একটি Program CPU এর নিয়ন্ত্রণে থাকে।

**প্রশ্ন-৩১. Multiprocessing System এর সুবিধা ও অসুবিধা লিখ।**

**উত্তর:** Multiprocessing System এর সুবিধা সমূহ:

- a) Parallel Processing এর জন্য কম্পিউটার এর পরামর্শদাতা বাড়ে।
- b) যদি একটি CPU নষ্ট হয়ে যায় তবে অন্যান্য CPU তাই Automatically এই CPU এর কাজ সম্পর্ক করে। কোন সিপিইউ সমস্যার কারণে কাজ ব্যবহৃত থাকে না।
- c) একই সাথে একাধিক জব প্রারম্ভ করা যায়।
- d) কাজের ব্যাপকতা সৃষ্টি হয়।

**Multiprocessing System এর অসুবিধা সমূহ:**

- a) Sophisticated Operating System এর প্রয়োজন হয়।
- b) বৃহৎ আকারের মেইন মেমোরি এর দরকার হয়।
- c) এই সিস্টেম ব্যায়বহুল হয় এবং রক্ষণাবেক্ষণ খরচ বেশি।

**প্রশ্ন-৩২. Time Sharing System এর সুবিধা ও অসুবিধা লিখ।**

**উত্তর:** Time Sharing System এর সুবিধা সমূহ:

- a) সিপিইউ এর Idle Time একেবারে কম।
- b) মুহূর্তের মধ্যে ফলাফল পাওয়া যায়।
- c) সব ব্যবহারকারী কম্পিউটার ব্যবহার করতে পারে।

d) একই সফটওয়্যার একাধিক ব্যবহারকারী ব্যবহার করতে পারে কলে ছাপ্পাকেট কপির প্রয়োজন হয় না।

e) Response Time খুবই কম হয়।

**Time Sharing System এর অসুবিধা সমূহ:**

- a) ডাটা নির্বাচন ও সংগঠনের ক্ষেত্রে সব সময় ব্যবহৃত হয়।
- b) আয়ৈ ডাটা Communication সমস্যা অন্তর্ভুক্ত হয়।
- c) একাধিক ব্যবহার করিগুলে সার্ভিস দিতে হয় বিধায় নির্ভর যোগ্যতা নির্দিষ্ট করা প্রয়োজন।
- d) যখন ব্যবহার করার সংখ্যা অতিরিক্ত ভাবে বেড়ে যায় তখন Result পেতে সময় লাগে।
- e) ডাটা সিকিউরিটি নির্দেশ করতে হয়।
- f) দামী অপারেটিং সিস্টেম দরকার।

**প্রশ্ন-৩৩. Time sharing system এর অপারেশন স্টেট সমূহ বর্ণন।**

**উত্তর:** Time Sharing System এর Operation State সমূহ: তাৎক্ষণিক মুহূর্তে একজন টাইম শেয়ারিং ব্যবহারকারী নির্দেশ যে কোন State এ থাকতে পারে।

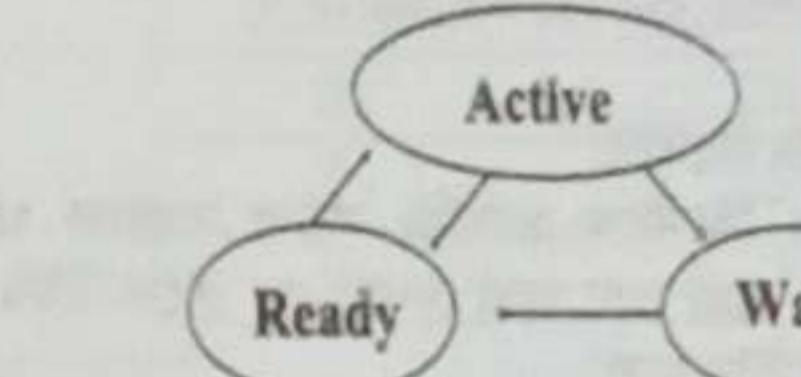


Figure: Block Diagram of Time Sharing System Operation

a) **Active:** User এর Program CPU ব্যবহার করছে। তখন User at a time active থাকতে পারে।

b) **Ready:** ব্যবহারকারীর প্রোগ্রাম Ready অর্থাৎ CPU ব্যবহার করতে যাচ্ছে।

c) **Wait:** ব্যবহারকারীর প্রোগ্রাম সিপিইউ ব্যবহার না করে I/O Operation এর জন্য অপেক্ষা করছে।

**প্রশ্ন-৩৪. Time Slice কি?**

উত্তর: টাইম শেয়ারিং ব্যবহৃত প্রত্যেক ব্যবহারকারীর জন্য একটি নির্দিষ্ট পরিমাণ সময় নির্ধারিত থাকে। এই সময় শেষ হলে অপর একটি প্রোগ্রামের জন্য টিক এ পরিমাণ সময় দেওয়া হয়। এই নির্ধারিত সময় অশেষে Time Slice বা টাইম স্লিপ বা কোয়ান্টাম বলে।

অন্যভাবে বলা যায়, Time Sharing System এ প্রত্যেক ব্যবহারকারী এর জন্য টাইম ভাগ করে দেওয়া থাকে অর্থাৎ এ টাইম পার হলে অন্য User Process করবে। এই টাইম এর ভাগকে Time Slice বলে।

**প্রশ্ন-৩৫. বিভিন্ন অপারেটিং সিস্টেমের মধ্যে তুলনা করন।** [NTRCA(Lecturer)-2019]

উত্তর: নিচে একার বিভিন্ন অপারেটিং সিস্টেমের মধ্যে ত

2. এই সিস্টেমে একটি প্রোগ্রাম শেষ হওয়ার পর পরবর্তী প্রোগ্রাম এক্সিয়াকশন করত হয়।
3. একটি কম্পিউটারে বলে কেবলমাত্র একটি প্রোগ্রাম নিয়েই কাজ করা যায়।
4. এ পদ্ধতিতে প্রোগ্রামের উপর তরক্কি প্রদান করা হয়।
5. উদাহরণ: এমএস ডস (MS-DOS)।
6. ব্যবহার: এ ধরনের অপারেটিং সিস্টেম ব্যাকিং সিস্টেম, বিভিন্ন ডিপার্টমেন্টেল স্টোরে ব্যবহার করা হয়।

#### রিয়েল টাইম অপারেটিং সিস্টেম:

1. যে অপারেটিং সিস্টেমে নির্বাচনের আদেশ প্রাপ্তির সাথে সাথে প্রোগ্রাম নির্বাচনের কাজ তরক্কি হয় তাকে রিয়েল টাইম অপারেটিং সিস্টেম বলে।
2. এই সিস্টেমে অ্যাগ্ল্যুটার ভিত্তিতে প্রোগ্রাম নির্বাচন করা হয়।
3. একই সাথে একাধিক টার্মিনালে বলে ভিন্ন ভিন্ন প্রোগ্রাম নিয়ে কাজ করা যায়।
4. এই সিস্টেমে প্রোগ্রামের উপর তরক্কি প্রদান করা হয়।
5. উদাহরণ: আরএমএস-৮৬ (RMS-86)।
6. ব্যবহার: এই ধরনের অপারেটিং সিস্টেম খাদ্য, গ্যাস, পেট্রোলিয়াম শিল্পে ব্যবহার করা হয়।

#### টাইম শেয়ারিং অপারেটিং সিস্টেম:

1. যে অপারেটিং সিস্টেমে প্রসেসিং করার সময়কে প্রোগ্রাম ও ব্যবহারকারীর মধ্যে ভাগ করে দেওয়া হয় তাকে টাইম শেয়ারিং অপারেটিং সিস্টেম বলে।
2. প্রতিটি প্রোগ্রাম নির্দিষ্ট সময় অক্ত অক্ত প্রসেসিং করা হয়।
3. একই সাথে একাধিক টার্মিনালে বলে ভিন্ন প্রোগ্রাম নিয়ে কাজ করা যায়।
4. এই সিস্টেমে সময়ের উপর তরক্কি প্রদান করা হয়।
5. উদাহরণ: ইউনিক্স (UNIX)।
6. ব্যবহার: নেটওর্কে অন লাইন ফাইল সিস্টেমে ব্যবহার করা হয়।

#### মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেম:

1. যে অপারেটিং সিস্টেমে একটি প্রসেসের সাথে সাথে একাধিক প্রোগ্রাম চালনা করা হয় তাকে মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেম বলে।
2. এই সিস্টেমে একই সময় একাধিক প্রোগ্রাম মেমরি করে এবং যখন একটি প্রোগ্রামের প্রসেসিং এর কাজ করে তখন পেরিফেরিয়াল ব্যবস্থা অন্য প্রোগ্রামের I/O Operation সম্পর্ক করে।
3. একটি কম্পিউটারে একটি প্রসেসের ব্যবহার করে একাধিক প্রোগ্রাম নিয়ে কাজ করা যায়।
4. এই সিস্টেমে প্রোগ্রাম সমূহের উপর তরক্কি প্রদান করা হয়।
5. উদাহরণ: উইন্ডোজ (Windows)।
6. ব্যবহার: এই ধরনের অপারেটিং সিস্টেম হোম ইউজাররা বেশি ব্যবহার করে।

#### মাল্টি প্রসেসিং অপারেটিং সিস্টেম:

1. যে সিস্টেমে একাধিক প্রসেসের মাধ্যমে এককে এক বা একাধিক প্রোগ্রামকে প্রসেসিং করা যায় তাকে মাল্টি প্রসেসিং অপারেটিং সিস্টেম বলে।

2. একেরে একসাথে একাধিক প্রোগ্রাম মেমোরিতে জমা থাকে। একই প্রোগ্রাম একাধিক প্রসেসের কিংবা ভিন্ন ভিন্ন প্রসেসেরে ভিন্ন প্রোগ্রাম প্রসেস করা হয়।
3. একটি কম্পিউটারে একাধিক প্রসেসের ব্যবহার করে একাধিক প্রোগ্রাম নিয়ে কাজ করা যায়।
4. এই সিস্টেমে প্রসেসের সমূহের উপর তরক্কি প্রদান করা হয়।
5. উদাহরণ: লিনাক্স (Linux)।
6. ব্যবহার: এই ধরনের অপারেটিং সিস্টেম মেইনফ্রেম ও আইএসপি তে বেশি ব্যবহার করা হয়।

#### প্রশ্ন-৩৬. Multiprocessing এর Requirements সমূহ উভয় Multiprocessing এর Requirements সমূহ:

- a) Large Memory
- b) Memory Protection
- c) Program Status Preservation
- d) Proper Job Mixing

#### প্রশ্ন-৩৭. Differentiate between multiprogramming and multiprocessing.

Ans: Multiprocessing refers to processing of multiple processes at same time by multiple CPUs. (একই সময়ে একাধিক কাজ অনেকগুলো সিপিইউ দিয়ে সম্পূর্ণ হবে)

Multiprogramming keeps several programs in main memory at the same time and execute them concurrently utilizing single CPU. (একই সময়ে একাধিক কাজ একটি কম্পিউটারে রেখে একটি সিপিইউ দিয়ে সম্পূর্ণ হবে)

#### প্রশ্ন-৩৮. What is the difference between time-sharing operating system and multi-programming operating system.

উত্তর: মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেম ও টাইম শেয়ারিং অপারেটিং সিস্টেম এর মধ্যে পার্থক্য নিম্নে দেওয়া হলো:

মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেম	টাইম শেয়ারিং অপারেটিং সিস্টেম
যে অপারেটিং সিস্টেমে একটি প্রসেসের সাথে সাথে একাধিক প্রোগ্রাম মেমরি করার সময়কে প্রোগ্রাম ও ব্যবহারকারীর মধ্যে ভাগ করে দেওয়া হয় তাকে মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেম বলে।	যে অপারেটিং সিস্টেমে একটি প্রসেসের সাথে সাথে একাধিক প্রোগ্রাম মেমরি করার সময়কে প্রোগ্রাম ও ব্যবহারকারীর মধ্যে ভাগ করে দেওয়া হয় তাকে টাইম শেয়ারিং অপারেটিং সিস্টেম বলে।
এই সিস্টেমে একই সময় একাধিক প্রোগ্রাম মেমরি করে এবং যখন একটি প্রোগ্রামের প্রসেসিং এর কাজ করে তখন পেরিফেরিয়াল ব্যবস্থা অন্য প্রোগ্রামের I/O Operation সম্পর্ক করে।	প্রতিটি প্রোগ্রাম নির্দিষ্ট সময় অক্ত অক্ত প্রসেসিং করা হয়।
এই সিস্টেমে একটি প্রসেসের ব্যবহার করে একাধিক প্রোগ্রাম নিয়ে কাজ করা যায়।	প্রতিটি প্রোগ্রাম প্রসেসিং করা যায়।
এই সিস্টেমে প্রোগ্রাম সমূহের উপর তরক্কি প্রদান করা হয়।	এই সিস্টেমে প্রোগ্রাম সমূহের উপর তরক্কি প্রদান করা হয়।

#### মাল্টি প্রসেসিং অপারেটিং সিস্টেম:

1. যে সিস্টেমে একাধিক প্রসেসের মাধ্যমে এককে এক বা একাধিক প্রোগ্রামকে প্রসেসিং করা যায় তাকে মাল্টি প্রসেসিং অপারেটিং সিস্টেম বলে।

তরক্কি প্রদান করা হয় :	প্রদান করা হয়।
উদাহরণ: উইন্ডোজ (Windows)।	উদাহরণ: ইউনিক্স (UNIX)।
ব্যবহার: এই ধরনের অপারেটিং সিস্টেম	ব্যবহার: নেটওর্কে অন লাইন ফাইল সিস্টেমে ব্যবহার করে।
হোম ইউজাররা বেশি ব্যবহার করা হয়।	ফাইল সিস্টেমে ব্যবহার করা হয়।

#### প্রটোকল লিখাত

প্রশ্ন-১. Partition কী?

উভয় প্রত্যেক Application software memory তে যে জায়গা সংরক্ষণ করে সে অংশটিকে Partition বলে।

#### প্রশ্ন-২. Foreground Program কী?

উভয় Multiprogramming system এ যে সকল Program এর Priority বেশি অর্থাৎ অপেক্ষাকৃত আছেই Execute হবে। সেই প্রোগ্রাম তালোকে Foreground Program বলে।

#### প্রশ্ন-৩. Background Program কী?

উভয় Multiprogramming System এ যে প্রোগ্রাম তালোর Priority অপেক্ষাকৃত কম অর্থাৎ দেরিতে এক্সিকিউট হয়। সেই প্রোগ্রাম তালোকে Background Program বলে।

#### প্রশ্ন-৪. Concurrent Process বলতে কি বুঝা?

উভয় Multiprogramming System এ একই সময়ে এবং একই সময়ে একাধিক প্রোগ্রাম কম্পিউটারের এর মেইন মেমোরিতে থাকে ও একত্রে প্রসেস হয়। একে Concurrent Process বলা হয়।

#### প্রশ্ন-৫. Time Sharing ব্যবহার কি ধরনের Facility দ্বারা?

উভয়:

- (i). Terminal
- (ii). Data Communication ব্যবহা
- (iii). Main Memory
- (iv). Secondary Memory

#### প্রশ্ন-৬. Multiprocessing System এ কি কি ধরনের প্রোগ্রাম থাকে? উভয় Multiprocessing System এ ২ ধরনের প্রোগ্রাম থাকে:

- a) I/O Bound Program
- b) CPU Bound Program

#### CPU Scheduling Algorithm

প্রশ্ন-১. Scheduling Algorithm তি লিখ এবং প্রত্যেকটি আলাদা আলাদা ব্যাখ্যা করুন। অথবা, List CPU Scheduling algorithms and their merits and demerits. [বিভিন্ন বর্ষান্ত-২০২০- সহকারী মেইন্টেনেন্স ইন্জিনিয়ার (নন-ক্যারার), ICT, বিভিন্ন বর্ষান্ত-২০১৭- সহকারী মেইন্টেনেন্স ইন্জিনিয়ার (নন-ক্যারার), ]

উভয় সিডিউলারের Operation সম্পর্ক করার জন্য বিভিন্ন ধরনের সিডিউল অ্যালগরিদম প্রচলিত আছে। যেমনঃ

- a) First Come First Serve Scheduling (FCFS)
- b) Shortest-Job-First Scheduling (SJFS)
- c) Priority Scheduling (PS)
- d) Round Robin Scheduling (RR)
- e) Multilevel Queue Scheduling (MQS)
- f) Multilevel Feedback Queue Scheduling (MFQS)

Formula:	Waiting time = Turnaround time - Burst time
অথবা	
Waiting time = In time- arrival time	

Turnaround time= Closing time- arrival time

</div

[মনে রাখবেন এখানে লক্ষণীয় যে Process এর Burst Time বেশি তার Processing Time বেশি লাগবে এবং এর Gantt Chart পর্ক হবে। প্রথমে যে প্রসেস Execute হবে তার Waiting Time 0 হতে থাকবে। পরে প্রসেস তালোর Waiting Time হবে পূর্বের ফিনিশ টাইম।]

Total Waiting Time = প্রতিটি প্রসেসের Waiting Time এর যোগফল =  $0 + 24 + 27 = 51 \text{ MS}$

$$\text{Average Waiting Time} = \frac{\text{Total waiting time}}{\text{Total Process}} = \frac{51}{3} = 17 \text{ MS} \quad (\text{Ans})$$

Total Turnaround Time = প্রতিটি Process এর Finishing Time এর যোগফল =  $24 + 27 + 30 = 81 \text{ MS}$

$$\text{Average Turnaround Time} = \frac{81}{3} = 27 \text{ MS} \quad (\text{Ans})$$

[N.B: Response Time = Waiting Time.]

উদাহরণ (২). নিম্ন পাঁচটি Process তাদের Arrival Time, Service Time (Burst Time) দেওয়া হল। Gantt Chart সৈথিক এবং Average Waiting Time & Average Turnaround Time বের করুন।

Process	Arrival time	Burst time/Service time
P <sub>1</sub>	1	8
P <sub>2</sub>	2	2
P <sub>3</sub>	3	1
P <sub>4</sub>	4	2
P <sub>5</sub>	5	5

Solution:

এখানে P<sub>1</sub>, P<sub>2</sub> যথাক্রমে Process 1, Process 2. যদি Process তালো P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub> এবং P<sub>5</sub> ক্রমানুসারে আসে এবং FCFS অনুসারে Serve করা হয়, তাহলে নিম্নের ন্যায় Gantt Chart অঙ্কন করা যেতে পারে।

Gantt chart:

P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
1	9	11	12	14 19

উপরের Gantt chart থেকে নিম্ন লিখিত Table তৈরী কর।

Process	Arrival Time	Finishing Time	Waiting Time	Turnaround Time =(Finishing Time-Arrival Time)
P <sub>1</sub>	1	9	0	(9-1)=8
P <sub>2</sub>	2	11	7	(11-2)=9
P <sub>3</sub>	3	12	8	(12-3)=9
P <sub>4</sub>	4	14	8	(14-4)=10
P <sub>5</sub>	5	19	9	(19-5)=14

প্রথম Process এর Waiting Time '0' হতে থাক হবে। আর যদি Arrival Time দেওয়া থাকে তাহলে বাকি প্রসেস তালোর Waiting Time হবে = (পূর্বের প্রসেস এর Finishing time-Arrival time)

$$\therefore P_2 = (9-2) = 7, P_3 = (11-3) = 8, P_4 = (12-4) = 8, \\ P_5 = (14-5) = 9$$

Total Waiting time = প্রতিটি প্রসেস এর Waiting time এর যোগফল =  $0 + 7 + 8 + 8 + 9 = 32 \text{ MS}$

$$\text{Average Waiting Time} = \frac{32}{5} = 6.4 \text{ MS}$$

$$\text{Total Turnaround Time} = 8 + 9 + 9 + 10 + 14 = 50 \text{ MS}$$

$$\text{Average Turnaround Time} = \frac{50}{5} = 10 \text{ MS. (Ans)}$$

ব্যাখ্যা:

(১). SJF: SJF algorithm এ যখন CPU Available থাকে তখন এই Process টি নির্বাচিত হবে যার Burst time সবচেয়ে কম। অর্থাৎ যে Process এর Execution time সবচেয়ে কম হয়েছে সেটিই প্রথমে Execute হবে। ইহা দুই ধরণের। যেহেন

- (1). Preemptive SJF and
- (2). Non Preemptive SJF

SJF এর সুবিধা:

- a) FCFS Algorithm এর ক্ষেত্রে Waiting time কম হ।
- b) Optimal পদ্ধতি।
- c) Possible User Abuse.
- d) একটি প্রসেস ক্রতৃপক্ষ তালো তা অবশ্যই জানতে হবে।
- e) Non-Preemptive Method.

SJF এর অসুবিধা:

- a) SJF এ starvation হয় যখন ছোট প্রসেস আসতে থাকে,
- b) short term CPU scheduling এর ক্ষেত্রে এটি implement করা যাবন।

উদাহরণ (৩). নিম্নে ৪ টি প্রসেস এবং তাদের Burst Time দেওয়া হল। SJF Algorithm ব্যবহার করে Average Waiting Time বের করুন। এবং Gantt Chart দেখান।

Process	Burst Time
P <sub>1</sub>	6
P <sub>2</sub>	8
P <sub>3</sub>	7
P <sub>4</sub>	3

Solution:

Gantt Chart: যেহেতু P<sub>4</sub> এর Burst Time কম আই P<sub>4</sub> Process টি আসে Execute হবে। তারপরে যথাক্রমে P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub> প্রসেস তালো নির্বাচিত হবে।

P <sub>4</sub>	P <sub>1</sub>	P <sub>3</sub>	P <sub>2</sub>
0 3	9	16	24

$$\text{Average Waiting Time} = \frac{0+3+9+16}{4} = 7 \text{ MS (Ans)}$$

Average Turnaround Time =  $\frac{3+9+16+24}{4} = 13 \text{ MS Ans}$

Response time (P<sub>n</sub>) = P<sub>n</sub> প্রসেস এর Waiting time - P<sub>n</sub> এর পূর্বে প্রসেসের Burst time.

$$\text{Response time (P<sub>1</sub>)} = 3-3=0$$

$$\text{Response time (P<sub>2</sub>)} = 16-7=9$$

$$\text{Response time (P<sub>3</sub>)} = 9-6=3$$

$$\text{Response time (P<sub>4</sub>)} = 0-0=0$$

$$\therefore \text{Average Response time} = \frac{0+9+3+0}{4} = 3 \text{ MS. (Ans)}$$

উদাহরণ (৪). Calculate The Average Waiting Time and Total Turnaround Time in following Process:

(1) Non Preemptive SJF and (2) Preemptive SJF

[সব প্রসেসের অধীনে জারীর সুবিধা প্রিফেরেন্স নাই।] ২০১১

Process      Arrival Time      Burst Time

P <sub>1</sub>	0	7
P <sub>2</sub>	2	4
P <sub>3</sub>	4	1
P <sub>4</sub>	5	4

Solution:

Non Preemptive SJF Gantt Chart:

P <sub>1</sub>	P <sub>3</sub>	P <sub>2</sub>	P <sub>4</sub>
0	7	8	12 16

Process	Waiting Time	Turnaround Time
P <sub>1</sub>	0	7-0=7
P <sub>2</sub>	8-2=6	12-2=10
P <sub>3</sub>	7-4=3	8-4=4
P <sub>4</sub>	12-5=7	16-5=11

$$\text{Total Waiting Time} = 0 + 6 + 3 + 7 = 16 \text{ MS}$$

$$\text{Average Waiting Time} = \frac{16}{4} = 4 \text{ MS (Ans)}$$

$$\text{Total Turnaround Time} = 7 + 10 + 4 + 11 = 32 \text{ MS}$$

$$\text{Average Turnaround Time} = \frac{32}{4} = 8 \text{ MS (Ans)}$$

$$\text{Throughput} = \frac{4 \text{ Jobs}}{16 \text{ Sec}} = 0.25 \text{ Jobs/Sec (Ans)}$$

Preemptive SJF Gantt Chart:

P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>2</sub>	P <sub>4</sub>	P <sub>1</sub>
0	2	4	5	7	11 16

[Preemptive এ যখন Process দুইবার আসবে তখন Waiting Time হবে পূর্বের Process এর (Start Time - Arrival Time) + (Start Time - Finish Time).]

$$\text{Waiting Time P}_1 = (0-0)+(11-2)=9$$

$$\text{Waiting Time P}_2 = (2-2)+(5-4)=1$$

$$\text{Total Waiting Time} = (9+1+0+2)=12$$

$$\text{Average Waiting Time} = \frac{12}{4} = 3 \text{ MS (Ans)}$$



- Ques**
- The Shortest Job First (SJF) has the highest priority.
  - The lowest level of Multilevel Feedback Queues (MLFQ) is First Come First Serve (FCFS).
  - First Come First Serve (FCFS) gives the highest priority to the job having been in existence the longest.
  - None

**Ans-2.** নিচেরিটি নিচেরিটি এলারিসমস এর মধ্যে short job এর পক্ষে বৈধযুক্ত (discriminate)- আসোচন করুন।

- FCFS
- RR
- Multi level feedback queues

**Ques**

a. FCFS: Discriminates against short jobs since any short jobs arriving after long jobs will have a longer waiting time.

b. RR: treats all jobs equally (giving them equal bursts of CPU time) so short jobs will be able to leave the system faster since they will finish first.

c. Multilevel feedback queues: work similar to the RR algorithm—they discriminate favorably toward short jobs.

**Ans-3.** Job arrival time and execution time of Operating system tasks table is given, [Rupali bank -ANE-2021]

Job	Arrival Time	Execution Time
A	0	10
B	3	7
C	5	3

find out average waiting time for FCFS, preemptive SJF, Round Robin (Quantum time: 3) scheduling algorithm.

**Answer:**

### FCFS

**Gantt Chart:**

A	B	C
0	10	17 20

**Order of service (table):**

Process	Arrival	Service	Priority	Start	Completion	turnaround time (TAT)	waiting time (WAT)
A	0	10	0	0	10	10	0
B	3	7	0	10	17	14	7
C	5	3	0	17	20	15	12
Avg:	-	6.666	-	-	-	13	6.333
		67				33	

### Preemptive SJF

**Gantt Chart:**

A	C	A	B
0	5	8	13 20

**Order of service (table):**

Process	Arrival	Service	Priority	Start	Completion	turnaround time (TAT)	waiting time (WAT)
C	5	3	0	5	8	3	0
A	0	10	0	0	13	13	3
B	3	7	0	13	20	17	10
Avg:	-	6.666	-	-	-	11	4.333
		67				33	

### Round Robin

**Gantt Chart:**

A	B	A	C	B	A	B	A
0	3	6	9	12	15	18	19 20

**Order of service (table):**

Process	Arrival	Service	Priority	Start	Completion	turnaround time (TAT)	waiting time (WAT)
C	5	3	0	9	12	7	4
B	3	7	0	3	19	16	9
A	0	10	0	0	20	20	10
Avg:	-	6.666	-	-	-	14.333	7.666
		67				3	67

**Ques-4.** What is the difference between preemptive and non preemptive scheduling. [Biju's question-2024]

সরকারী যোগাযোগ (সম.-ক্লাশের) :

উত্তর: মিসেস প্ৰেমিত এবং non preemptive scheduling এৰ মাঝে পাৰ্থক্য দেওয়া হৈলো:

Preemptive scheduling	Non-preemptive scheduling
CPU প্ৰসেসৰ জন্য নিমিট সময় allocate কৰে।	CPU প্ৰসেসৰ জন্য পৰ্যট সময় allocate কৰে।
Priority এৰ উপৰ ভিত্তি কৰে কৱে switch কৰে।	switch কৰে না।
Process ready state থেকে running state & switching এ সহজ লাগে।	switching এ সহজ লাগে।
Flexible	Flexible না।

There is a tie between the priorities of P1 and P3. If P1 is scheduled first:

P2	P5	P1	P3	P4	Priority13
0	1	6	16	18	19

Or if for the tie between P1 and P3 priorities, P3 is scheduled first:

P2	P5	P3	P1	P4	Priority31
0	1	6	8	18	19

P1	P	P	P	P	P	P	P	P	P	R
0	1	2	3	4	5	6	7	8	9	1

Answer: b. Turnaround Times

b. The turnaround time of each process for each of the scheduling algorithms in part a:

$$\text{Turnaround\_time} = \text{Finish\_time} - \text{Arrival\_time}$$

FCFS	SJF	Priority1	Priority3	RR
P1	10	19	16	18
P2	11	1	1	1
P3	13	4	18	8
P4	14	2	19	19
P5	19	9	6	6

Answer: c. Waiting Times:

c. The waiting time of each process for each of the scheduling algorithms in part a:

$$\text{Waiting time} = \text{Starting time} - \text{Arrival time}$$

FCFS	SJF	Priority13	Priority31	RR
P1	0	9	6	8
P2	10	0	0	1
P3	11	2	16	6
P4	13	1	18	3
P5	14	4	1	9

Answer: d. Smallest Average Waiting Time:

d. The schedule in part that results in the minimal average waiting time (over all processes):

	FCFS	SJF	Priority1	Priority3	RR
Ave. wait	9.6	3.2	8.2	6.6	5.4

So the answer is Shortest Job First.

**প্রশ্ন-১.** নিম্ন সর্টিং Process এর তালের Burst time দেওয়া আছে। Time Quantum 2 MS। RR Scheduling Algorithm অনুসারে Average Waiting time এবং Average Response time কেন করুন।

Process	Burst time
P <sub>1</sub>	16
P <sub>2</sub>	10
P <sub>3</sub>	8
P <sub>4</sub>	13

**প্রশ্ন-২.** নিম্ন সর্টিং Process এর তালের CPU Burst time দেওয়া হচ্ছে। FCFS, SJF & RR (Quantum time-3) Algorithm অনুসারে Average waiting time এবং Turnaround time, Average response time কেন করুন।

Process	Burst time
P <sub>1</sub>	10
P <sub>2</sub>	29
P <sub>3</sub>	3
P <sub>4</sub>	7
P <sub>5</sub>	12

**প্রশ্ন-৩.** নিম্ন সর্টিং Process এর তালের Arrival time & Burst time দেওয়া আছে। FCFS, SJF & RR (Quantum time-3) Algorithm অনুসারে Average Waiting time, Turnaround time এবং Average Response time কেন করুন।

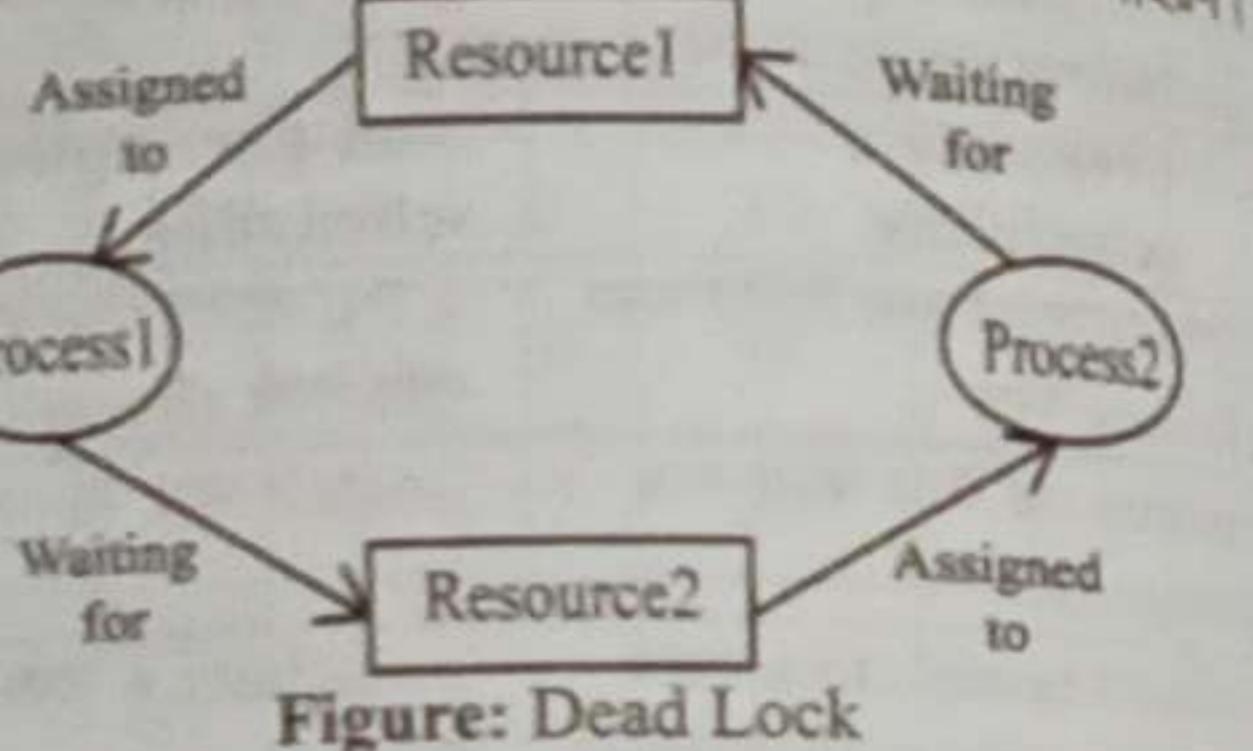
Process	Arrival time	Burst time
P <sub>1</sub>	0.0	8
P <sub>2</sub>	0.4	4
P <sub>3</sub>	1.0	1

### Deadlock Handling Methods

**প্রশ্ন-৪.** Deadlock কাকে বলে? [INTRCA(Lecturer)-2014 & 2016, মিলি মুন্ডার-২০২০, সহকারী যোগাযোগ ইতিবাচক (মন-ক্যাভার) (ICT), 38<sup>th</sup> BCS, মন্তব্য কলাপ ও বৈদেশিক কর্মসূচন কর্মসূচি কর্মসূচি বৃক্ষে-২০১৮, মিলি মুন্ডার-২০১৫, সহকারী যোগাযোগ ইতিবাচক (মন-ক্যাভার)]

উভয় Multi Programming System এ কখনো কখনো এমন অবস্থা হত যখন একটি Process কার্যত অবস্থার অন্য একটি Process এ Request প্রার্থ কিন্তু পূর্বে Processing কর Process টির Execution সম্পর্ক ন হওয়া পর্যন্ত পরবর্তী Process Processing এ যেতে পারে না। এই অবস্থায় সে Waiting State এ থাকে। Waiting

State এ থাকার এই Condition কেই Dead Lock বলে। যেমন একটি সকল গুলিতে (যে গুলিতে সর্বোচ্চ একটি বিজ্ঞ তালে পারে) সৃষ্টি নির থেকে সৃষ্টি বিজ্ঞ আসতে থাকলে Deadlock এর সৃষ্টি হয়। আছড়াও নদী পারা পার, রেললাইন বাত্তর Deadlock এর উদাহরণ।



**প্রশ্ন-৫.** Deadlock সৃষ্টির কারণ বা সংগঠনের শর্তবর্তী কি কি? [INTRCA(Lecturer)-2014, মিলি মুন্ডার-২০২০- সহকারী যোগাযোগ ইতিবাচক (মন-ক্যাভার) (ICT), বিভিন্ন মহাশয়-২০১৯-সহকারী যোগাযোগ (মন-ক্যাভার), আর যোগাযোগ ও প্রতিক বিজ্ঞ-২০১৯-জাতি এন্ট্রি স্টুডিওজন, Instructor ME WOE -2018] বা, একটি System Deadlock এ পড়বার শর্তবর্তী করুন।

উভয় Deadlock সৃষ্টির কারণ বা সংগঠনের শর্তবর্তী:

(i). Mutual Exclusion: যদি কোন প্রসেস একটি রিসোর্স কে যেত (Hold) করে রাখার প্রয়োগ অন্য রিসোর্স ব্যবহার করতে চায়। তখন Deadlock Condition সৃষ্টি হবে।

(ii). Hold and Wait: যখন কোন প্রসেস একটি রিসোর্স কে যেত (Hold) করে রাখার প্রয়োগ অন্য রিসোর্স ব্যবহার করতে চায়। তখন Deadlock Condition সৃষ্টি হবে।

(iii). No-Premition: যে সকল রিসোর্স পূর্বৰ্তী কোন প্রসেস এর জন্য Allocate করা হয়েছে, সে সকল রিসোর্স এর জন্য কোন প্রসেস কর্তৃক বিকেয়েট পাঠানো হলে তেওঁ শুধু অবস্থার সৃষ্টি হবে।

(iv). Circular Wait: অক্ষেত্রে এক সেট প্রসেস Waiting State এ থাকে। যদি এসেস Set{P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>} এর P<sub>0</sub>, P<sub>1</sub> থেকে যেকোনো রিসোর্স এর জন্য P<sub>1</sub>, P<sub>2</sub> থাকা যেকোনো রিসোর্স এর জন্য এবং P<sub>2</sub>, P<sub>3</sub> থেকে যেকোনো রিসোর্স এর জন্য Wait করতে থাকে তবে সেকেরে Deadlock অবস্থার সৃষ্টি হবে।

**প্রশ্ন-৬.** Deadlock এভিয়েরের উপায় কী কি? [মন্তব্য কলাপ]

উভয় Deadlock এভিয়েরের উপায় কী কি?

(i). Mutual Exclusion: Computer Operation এর সময়ে Non-Shareable Resource ব্যবহার না করা।

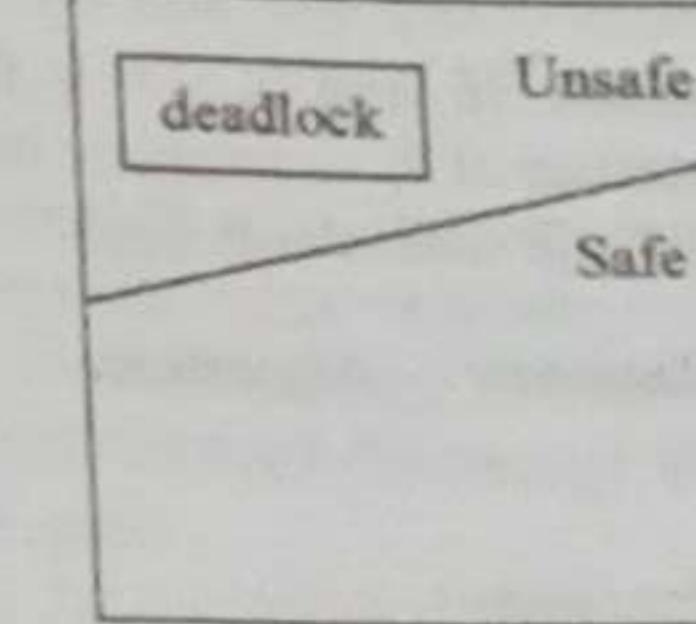
(ii). Hold and Wait: সিস্টেমে এমন ব্যবহাৰ রাখতে হবে যে কোন প্রসেস যদি কোন রিসোর্স কে যেত (Hold) করে তবে অন্য কোন প্রসেস এর জন্য বিকেয়েট পাঠানো হবে।

(iii). No-Premition: যে সকল রিসোর্স পূর্বৰ্তী কোন প্রসেস এর জন্য Allocate করা হবে। অন্যান্য প্রসেসকে অবশ্যই অপেক্ষ করতে হবে যে পর্যন্ত Allocated Resource উপরে অন্য প্রসেস (Process) যুক্ত (Free) না করে। যদ্যপি যেকোন কোন সিস্টেম জটিত প্রসেসের জন্য Resource ব্যবহাৰ করতে পারে, তখনই তাকে "Safe State" বলে ধৰা হবে। যদ্যপি যেকোন সিস্টেমটি "Safe State" এ থাকবে, ততক্ষণ Deadlock

হতে Release না হওয়া পর্যন্ত নতুন কোন প্রসেস হতে বিকেয়েট ন পাঠানো।

(iv). Circular Wait: সার্কুলের ওয়েট প্রতিবেদে এমন ব্যবহাৰ হৈন কৰা যাতে প্রতিটি প্রসেস Enumeration (সংখ্যাগুলো বা সমন্বয়) প্রক্রিয়া Increasing Order এ Resource এর জন্য বিকেয়েট পাঠানো পারে।

পরিহার কৰা যাবে, আবার State Unsafe হলৈ Deadlock সৃষ্টি হবে এমন কোন যুক্ত নেই।



**প্রশ্ন-৭.** Deadlock মোকাবেলা কৰার পদ্ধতি সমূহ লিখো।

উভয় Deadlock মোকাবেলা কৰার পদ্ধতি সমূহ:

- ডেভলক মোকাবেলা কৰার জন্য আমৰা এমন প্রয়োকল ব্যবহাৰ কৰতে পারি যেন সিস্টেমে কোন Dead Lock ন আসে।
- সিস্টেমে ডেভলক অবহাৰ কৰে, তিহিত কৰে এবং তা পুনৰুজ্জীব কৰে।
- আমৰা সবক্ষেত্ৰে ডেভলক সমস্যা অবহোলা কৰতে পারি।

উভয় Deadlock মোকাবেলা কৰার পদ্ধতি সমূহ:

Safety Algorithm: Deadlock Avoidance Safe State Algorithm এবি, একটি সিস্টেমে 12 টি Tape Drive & 3 টি Process বিবৃত্যান। Process P<sub>0</sub> এর জন্য 10 টি, P<sub>1</sub> এর জন্য 4 টি এবং P<sub>2</sub> এর জন্য 9 টি Tape Drive প্রয়োজন হতে পারে। T<sub>0</sub> সময়ে P<sub>0</sub> 5 টি, P<sub>1</sub> 2 টি, P<sub>2</sub> 2 টি Tape Drive hold কৰে আছে। এমতাৰজৰ জন্য আমৰা সবক্ষেত্ৰে ডেভলক সমস্যা অবহোলা কৰতে পারি।

Request কৰে যেন একটি প্রয়োকল পৰিবৰ্তিত হৈন।

Deadlock Avoidance (Deadlock Avoidance):

(1) Let, Work and Finish be vectors of length m and n.

Initialize,

Work = Available and

Finish[i] = false for i=0, 1, 2, 3... n

(2) Find and i such that both

a. Finish[i] =false

b. Need i ≤ work

If no such i exists, go to step 4.

(3) Work = Work + Allocation.

Finish[i] = true.

Go to step 2.

(4) If Finish[i] = true for all i, then the system is in a Safe State. This algorithm may require an order of m × n<sup>2</sup> operation to determine whether a state is safe.

**Safe State Algorithm:** Deadlock Problem Avoid কৰার জন্য Scheduling Algorithm হৈলো Banker's Algorithm. যখন কোন নতুন প্রসেস (Process) সিস্টেমে প্রবেশ (Enter) কৰে, তখন সেই প্রসেসকে অবশ্যই আৰ অকোক বিসোৰেও (Resource) জন্য প্রয়োজনীয় স্বৰূপক রিসোৰ্স (Resources Declare) কৰে। পৰবৰ্তীতে প্রসেসের Priority (অলগোরিদম) অনুযায়ী রিসোৰ্স সমূহকে Information অদান কৰে। একাবে প্রতিকৃত অ্যালগোরিদম নিয়ন্তা অদান কৰে যে, সিস্টেমটি Deadlock সিস্টেমে প্রবেশ কৰবে না। Deadlock পৰিহাৰ Algorithm কৰলোৱ মধ্যে Safe State Algorithm, Resources Allocation Graph Algorithm, Banker's Algorithm অন্যতম।

**Banker's Algorithm:** Banker's Algorithm বাবুৰামনেৰ জন্য স্থিলিত Data Structures অবশ্যই Maintain (পালন কৰা) কৰতে হবে।

**Data Structures:**

Let, n= Number of Processes, and m= Number of Resources types.

**Available:** Vector of Length m, If available [j] =k, there are k instances of resource type R<sub>j</sub> available.

**Max:** n × m Matrix, If Max [i, j] =k, then Process P<sub>i</sub> may request at most k instances of resource type R<sub>j</sub>.

**Allocation:**  $n \times m$  matrix. If allocation  $[i, j] = k$ , then  $P_i$  is currently allocated  $k$  instances of  $R_j$ .

**Need:**  $n \times m$  matrix. If need  $[i, j] = k$ , then  $P_i$  may need  $k$  more instances of  $R_j$  to complete its task.

Need  $[i, j] = \text{Max}[i, j] - \text{Allocation}[i, j]$

#### Resources Request Algorithm (Banker's Algorithm):

Let,

$P_i$  = Number of Processes

$R_j$  = Number of Resources

$K$  = Request Demanded by Process

যদি Request  $i(j) = k$  হয়, তখন Process  $P_i$ , Resource  $R[j]$  এর জন্য  $k$  সংখ্যক Requests পাঠাবে। যতক্ষণ পর্যন্ত না Process তার সর্বোচ্চ দাবি (Claims) নির্বাহ (Executed) করতে না পারে।

(1) If  $\text{Request}_i \leq \text{Need}_i$ ; Go to Step 2. অন্যথায় তুল (Error) সংবেদিত হবে।

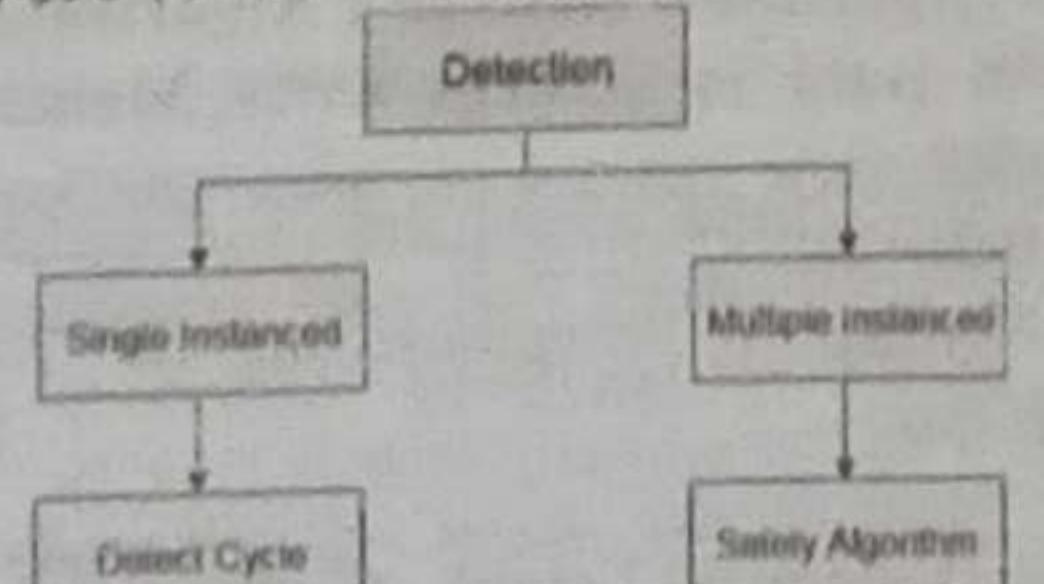
(2) If  $\text{Request}_i \leq \text{Available}$ , Go to Step 3, অন্যথায় Process  $P_i$  Waiting এ থাকবে। (যতক্ষণ পর্যন্ত না Resources Available না হয়।)

(3) Available = Available - Request;  
Allocation<sub>i</sub> = Allocation<sub>i</sub> + Request;  
Need<sub>i</sub> = Need<sub>i</sub> - Request;

যদি Result টি Safe State এর হয়, তাহলে Process  $P_i$ , Resource সমূহে Allocating হবে, অন্যথায়  $P_i$  অপেক্ষা করবে।

#### প্রশ্ন-৬. Describe the way of detecting and protecting deadlock. [38<sup>th</sup> BCS]

ডেভলক ডিটেকশন: OS অচল অবস্থা এড়াতে বা প্রতিরোধ করার জন্য কোন mechanism প্রয়োগ করে না। সুতরাং সিস্টেমটি বিবেচনা করে যে অচলাবস্থা অবশ্যই ঘটবে। অচলাবস্থা থেকে মুক্ত পাওয়ার জন্য OS পর্যায়বর্তনে যে কোনও অচলাবস্থার জন্য সিস্টেমটি পরিষ্কার করে। সেক্ষেত্রে এটি অচলাবস্থার সঞ্চালন করে তারপরে OS কিছু recover কৌশল ব্যবহার করে সিস্টেমটি recover করবে। OS এর মূল কাজটি হলো ডেভলকগুলি সনাত্ত করা। Resource allocation graph এর সাহায্যে OS ডেভলক সনাত্ত করতে পারে।



Single instanced resource types এ, যদি সিস্টেমে একটি cycle গঠিত হয় তবে অবশ্যই একটি deadlock থাকবে। অন্যদিকে, multiple instanced resource type থাকে, একটি cycle সনাত্ত করা যথেষ্ট নয়। resource allocation graph কে allocation যান্ত্রিক এবং request যান্ত্রিকে উপর্যুক্ত করে।

প্রশ্ন-৭. Deadlock সনাত্তের Algorithm শিখুন এবং করুন করুন। উত্তর যদি কোন সিস্টেমে Deadlock Prevention বা Deadlock Avoidance Algorithm প্রয়োগ না করা থাকে, তবে Deadlock Situation তৈরি হতে পারে। তাই প্রথমত দুটি কাজ করা দরকার যেহেন:

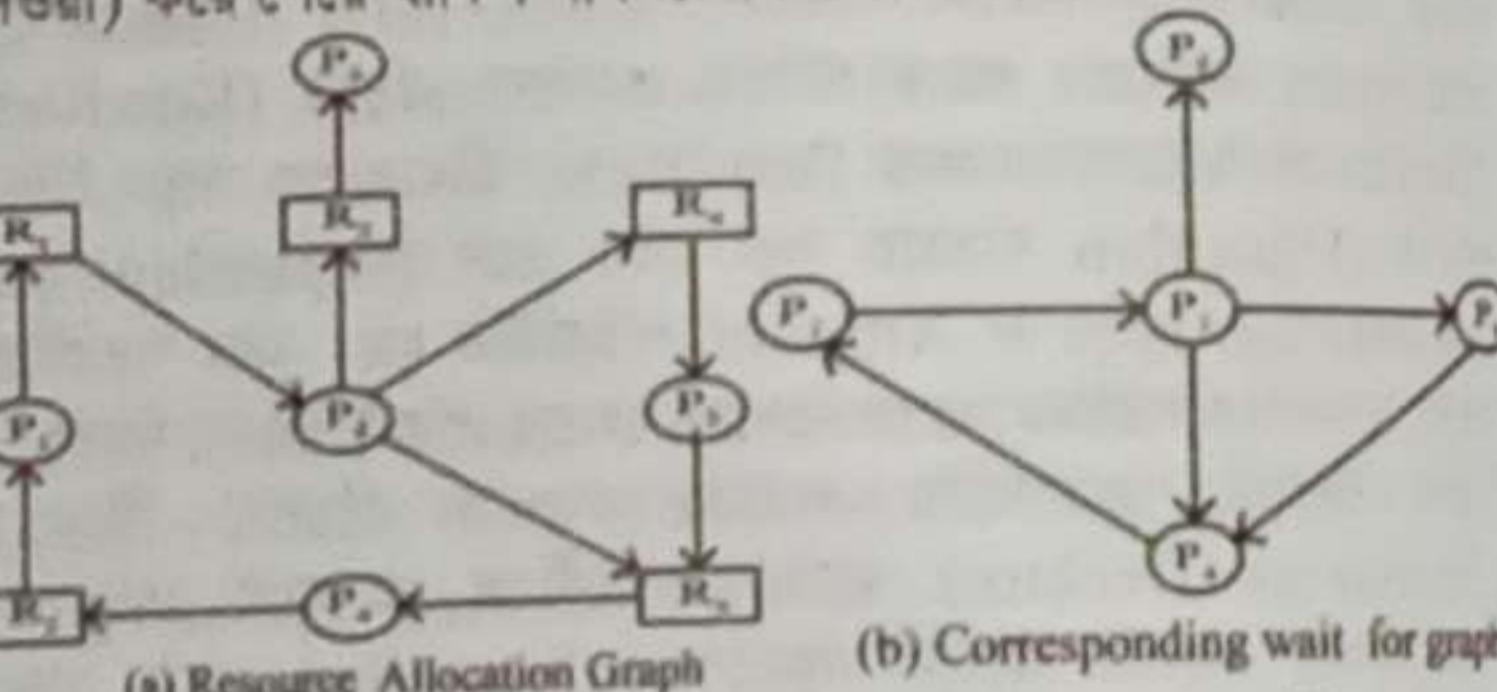
- a) System- এ Deadlock Situation তৈরি হলো কিনা আবৃজ্জে বের করা। এবং
- b) Deadlock Situation ঘটে থাকলে তা Recover (পুনরুজ্জীবন) করা।

সাধারণত Deadlock সনাত্তের Algorithm ২ টি। যেহেন:

- (i). Single Instance of Each Resource Type এবং
- (ii). Several Instance of a Resource Type

#### (i). Single Instance of Each Resource Type

**Algorithm:** যদি প্রতিটি রিসোর্সের (Resources) ক্ষেত্রে কোন একটি Instance থাকে তাহলেই আমরা এখন একধরনের Deadlock Detection Algorithm Define করতে পারি। যা কিনা "Wait for Graph" ব্যবহার করে। এইক্ষেত্রে আমরা এ গাফটিক Resource Allocation Graph থেকে Node (নোড) গুলো কান দিয়ে এবং প্রয়োজনীয় Edge (ধারা) সমূহ Collapse (পতন বা তেজে দেওয়া) করে পেয়ে থাকি। যা নিচের চিত্রে দেখানো হলো-



#### (ii). Several Instance of a Resource Type

**Algorithm:** যখন কোন Resource টাইপের জন্য একাধিক Instance প্রয়োগ হয়, তখন সেক্ষেত্রে তিনি ধরণের Deadlock Detection Algorithm ব্যবহার করা হয়। নিচে সেটি তুলে থাকলো-

##### Algorithm:

Let,

Work = Vectors of Length m

Finish = Vectors of Length n

##### Step (1): Initialize:

Work = Available, For  $i = 1, 2, \dots, n$ ,  
if Allocation  $i \neq 0$ ,

then,

Finish<sub>i</sub> = false;

Otherwise,

Finish<sub>i</sub> = true.

##### Step (2): Find an index i such that both:

- (a) Finish<sub>i</sub> == false
- (b) Request<sub>i</sub> ≤ Work
- If no such i exists,  
Go to step 4.

Step (3): Work = Work + Allocation;

Finish<sub>i</sub> = true

Go to step 2.

Step (4): If Finish<sub>i</sub> == false, for some  $i$ ,  
 $1 \leq i \leq n$ ,

Then,  
The system is in deadlock state.

Moreover, if Finish<sub>i</sub> == false,

Then,  
 $P_i$  is deadlocked.

[Algorithm requires an order of  $O(m \times n^2)$  operations  
to detect whether the system is in deadlocked state.]

#### প্রশ্ন-৮. Resources (রিসোর্স) কাকে বলে? ইহার প্রকারভেদ আলোচনা করুন।

উত্তর: রিসোর্স হলো এমনকিছু, যা কোনো একটি প্রসেস কোন নির্দিষ্ট সময়ে ব্যবহার করতে পারে। রিসোর্স হতে পারে হার্ডওয়ার (যেমন ট্যাপ ড্রাইভ, মেমোরি, হার্ডডিক ইত্যাদি) বা কোনো Information ইত্যাদি। বাস্তবে অপারেটিং সিস্টেম একটি রিসোর্স ম্যানেজার। রিসোর্সকে সাধারণত চারভাগে ভাগ করা যায়। যেহেন:

- a) Preemptable Resources
- b) Non-Preemptable Resources
- c) Shared Resources
- d) Dedicated Resources

নিচে একলো সম্পর্কে আলোচনা করা হলো।

(i) **Preemptable Resources:** যখন কোন রিসোর্স প্রসেস এ থাকে অবস্থা Priority (আধিকার) এর ভিত্তিতে অন্য প্রসেসের Execution এ সম্পর্ক করে, তখন তাকে Preemptable Resources বলে। যেহেন: মেমোরি এক প্রকার Preemptable Resources।

(ii) **Non-Preemptable Resources:** Process Running অবস্থায় যদি কোন রিসোর্স এর জন্য Request করা হয় এবং এই রিসোর্স এ বিদ্যমান প্রসেসের নির্বাহ সম্পর্ক না হওয়া পর্যন্ত রিসোর্সটি Free (মুক্তি) না হয়, তখন সেইসব রিসোর্সকে Non-Preemptable Resources বলে। যেহেন: প্রিটার, পেন ড্রাইভ ইত্যাদি।

(iii) **Shared Resources:** যখন কোন রিসোর্সকে অনেক গুলো প্রসেস থাকে Simultaneously (একসঙ্গে) ব্যবহার করা যায়, তখন তাকে Shared Resources বলে।

(iv) **Dedicated Resources:** যখন কোন রিসোর্স কুমোর নির্দিষ্ট কোন প্রসেস এর জন্য ব্যবহার করা থাকে।

প্রশ্ন-৯. ডেভলক থেকে মুক্ত পাওয়ার পক্ষত আলোচনা করুন। উত্তর: ডেভলক থেকে মুক্ত পাওয়ার পক্ষত: যখন কোন সিস্টেমে ডেভলক অবস্থার সৃষ্টি হয়, তখন তা Recovery বা তা থেকে মুক্ত পাওয়া অভ্যর্থনা। একেরে ডেভলক ভাসার (Breaking) দুই ধরণের পক্ষত বা Option রয়েছে। যেহেন:

- (a) Aborting Process Method
- (b) Resources Prompt Method

#### (c) Aborting Process Method আবার দুই প্রকার। যেহেন:

(i) **Abort All Deadlock Process:** এই পক্ষতে সম্পূর্ণ ডেভলক Cycle টিকে তেজে দেওয়া হয়। যার ফলে নীর্ধারিত ধরে প্রসেস থাকা Computed বিভিন্ন রেজিস্ট সমূহ নষ্ট হয়ে যায়। ফলে পরবর্তীতে Recompute করা লাগে।

#### (ii) Abort One Process at a time Until the Deadlock Cycle is Eliminated:

এই পক্ষতে একটি করে প্রসেস Abort করা হয় এবং যতক্ষণ পর্যন্ত ডেভলক অবস্থার সিস্টেমটি অবস্থান করে ততক্ষণ পর্যন্ত এ প্রক্রিয়া চলতে থাকে।

(d) **Resources Prompt Method** এ ডেভলক Cycle এর এক পক্ষতে একাধিক রিসোর্সের Preemption ঘটিয়ে ডেভলক Recovery করা যায়। এইক্ষেত্রে তিনটি বিদ্যমান লক্ষ্যনীয়। যেহেন:

- (i) Selecting a Victim: সবার আগে সঠিক রিসোর্স বা প্রসেসটি Select করে Preemption ঘটানো, ফলে এতে ব্যর্থ ও সময় করা লাগে।

(ii) **Roll Back:** যেকোন প্রসেস বা রিসোর্স Preempt করার পর উক্ত রিসোর্স বা প্রসেসকে Normal Execution এর উদ্দেশ্যে Safe (নিরাপদ) State এ নিয়ে যাওয়া এবং তাকে এ State থেকে করে তার Reusability (পুনর্ব্যবহার) নির্ভিত করা।

(iii) **Starvation:** যখন কোন সিস্টেমের Cost Factor বিবেচনা করে Victim Select করে Preempt করা হয় এবং তাকে Roll Back করা হয়, তখন এই Victim বাস্তবে ডেভলক সৃষ্টি করতে পারে। তাই নির্দিষ্ট সংখ্যাকার পর্যন্ত Victim কে Roll Back করার পর তাকে করে Damage দেওয়াই হচ্ছে Starvation।

#### প্রশ্ন-১০. কম্পিউটার সিস্টেমের Environment (পরিবেশ)

ব্যাক্তিরেকে ডেভলক এর তিনটি বাস্তব উদাহরণ শিখুন।

উত্তর: কম্পিউটার সিস্টেমের Environment (পরিবেশ) ব্যাক্তিরেকে ডেভলক এর তিনটি বাস্তব উদাহরণ। যেহেন:

- a) একটি সিঙ্গেল লেন ট্রীজেল উপর দিয়ে দুটি গাড়ি (কার) এর বিপরীত দিক থেকে যাওয়া আসা। (Two cars crossing a single-lane bridge from opposite directions.)

- b) একজন ব্যক্তির একটি ল্যাডার দিয়ে নামতে যাওয়া যখন অন্যজন সেই ল্যাডার দিয়ে উপরে উটতে চার। (A person going down a ladder while another person is climbing up the ladder.)

- c) দুটি ট্রেনের একই রেললাইনের মধ্য দিয়ে একে অপরের দিকে আসা। (Two trains traveling toward each other on the same track.)

প্রশ্ন-১১. ধরুন, একটি সিস্টেম একই টাইপের ৪ টি রিসোর্সকে ৩ টি প্রসেস দ্বারা শেয়ার করেছে। যেখানে কিনা প্রত্যেক প্রসেসের ২ টি করে রিসোর্স প্রয়োজন। সিস্টেমটি ডেভলকড: যার দ্বারা বৃক্ষ যাজে যে, প্রত্যেক প্রসেসই একটি করে রিসোর্স হোল্ড করে আছে এবং প্রত্যেক প্রসেসই আরো একটি রিসোর্স হোল্ড করার জন্য অপেক্ষা করছে। যেহেতু এখানে ৩ টি প্রসেস ও ৪ টি রিসোর্স, তাই বলাই যাচ্ছে যে, যেকোন একটি প্রসেস ২ টি রিসোর্স হোল্ড করতে পারবে এবং এ প্রসেসের অন্য কোন রিসোর্স হোল্ড করার প্রয়োজন নেই এবং, অতএব ইহা তার রিসোর্সের কাছে পুনরায় ফিরে আসতে পারবে যখন তার প্রসেসিং শেষ হবে।

প্রশ্ন-১২. নিম্নোক্ত Resource-Allocation Policy বিবেচনা করুন।  
১) রিসোর্সের এর জন্য Requests এবং Releases যেকোন সময় বরাদ্দ দেওয়া যায়। যদি রিসোর্সসকে একটি Request সংরক্ষণ করতে না পারে, তার কারণ হচ্ছে রিসোর্স Available (সহজলভ) না, তখন আমরা Blocked Process ডলো চেক করি, যেতে কিনা রিসোর্সের এর জন্য অপেক্ষা করছে। যদি Desired Resources থাকে, তখন এই রিসোর্স এখান থেকে তুলে নেওয়া হয় এবং তাদেরকে Requesting Process দেওয়া হয়। অপেক্ষারত প্রসেস এর জন্য রিসোর্সের এর ভেঙ্গের সেই বৃক্ষ প্রাপ্ত অপেক্ষা যে রিসোর্সসকে তুলে নেওয়া হয় তার সাথে অন্তর্ভুক্ত করা হয়।

উদাহরণ স্বরূপ, তিন ধরণের রিসোর্স সহকারে একটি সিস্টেম বিবেচনা করুন এবং ভেঙ্গের ভলোর Available initialized (৪,২,২)। যদি  $P_0$  প্রসেস (২,২,১) এর জন্য জিজ্ঞাসা করে, ইহা তাদেরকে পার। যদি  $P_1$  (১,০,১) এর জন্য জিজ্ঞাসা করে, ইহা তাদেরকে পার। তখন যদি  $P_0$  প্রসেস (০,০,১) এর জন্য জিজ্ঞাসা করে, ইহা তখন ব্লকড হবে যায় (resource not available)। এখন যদি  $P_2$  (২,০,০) এর জন্য জিজ্ঞাসা করে, ইহা Available (১,০,০) কে পাবে এবং আর একটি পাবে যা  $P_0$  এর জন্য বরাদ্দ ছিলো (Since  $P_0$  is Blocked)।  $P_0$ 's Allocation এর vector (১,২,১) এ চলে যাবে এবং ইহার ভেঙ্গের (১,০,১) তে চলে যাওয়া প্রয়োজন।

(ক) এখানে কি ডেভলক ঘটতে পারে? যদি ঘটে, উদাহরণ দিন। যদি না ঘটে, কোন প্রয়োজনীয় ক্ষিণনের জন্য ঘটতে পারেনি, শিখুন।

(খ) অনিদিষ্ট ব্লকিং ঘটতে পারে?

উত্তরঃ

(ক) Deadlock cannot occur because preemption exists.

(খ) হ্যাঁ, অনিদিষ্ট ব্লকিং ঘটতে পারে। কারণ একটি প্রসেস তার প্রয়োজনীয় সব রিসোর্সে কথনো অর্জন করতে পারেনা যদি তারা একটি

সিরিজের দ্বারা নিয়ন্ত্রিত Preempted হয়ে থাকে যেমন যাদের মধ্যে একটি Process C।

প্রশ্ন-১৩. এমন একটি সিস্টেম, যেখানে  $\leq$  টি Process ( $P_0, P_1, P_2, P_3$ , এবং  $P_4$ ) রয়েছে এবং  $\leq$  টি Resource type রয়েছে। Resource type  $r_1$  এর ১০ টি Instances রয়েছে, Resource type  $r_2$  এর ৫ টি ও Resource type  $r_3$  এর ৭ টি Instances রয়েছে। নিম্নে  $T_0$  সময়ে Resource State দেখানো হলো-

Process	Allocation			Available
	$r_1$	$r_2$	$r_3$	
$P_0$	0	1	0	7 5 3
$P_1$	2	0	0	3 2 2
$P_2$	3	0	3	9 0 2
$P_3$	2	1	1	2 2 2
$P_4$	0	0	2	4 3 3

যাকোর্স এলগরিদম অনুসৰে-

- ১) Need Matrix কী হবে?
- ২) উন্নিষিত সিস্টেম কী Safe State এ যাবে?

সমাধান:

- ১) Need Matrix এর ডেটা নিম্নরূপ হবে-

Process	Need		
	$r_1$	$r_2$	$r_3$
$P_0$	7	4	3
$P_1$	1	2	2
$P_2$	6	0	0
$P_3$	0	1	1
$P_4$	4	3	1

- ২) Process  $P_0$  এর জন্য:

Work ( $P_0$ ) = Current Available = 332

Need ( $P_0$ ) = 743

Need ( $P_0$ )  $\leq$  Work ( $P_0$ )

$743 \leq 332$  is False

So Finish ( $P_0$ ) = False.

(যেহেতু Need এবং Work কম, সুতরাং Finish হবেন।)

- ৩) Process  $P_1$  এর জন্য:

Work ( $P_1$ ) = Current Available = 332

Need ( $P_1$ ) = 122

Need ( $P_1$ )  $\leq$  Work ( $P_1$ )

$122 \leq 332$ , is True

So Finish ( $P_1$ ) = True.

Current Work, Work = Work + Allocation ( $P_1$ )  
 $= 332 + 200 = 532$

- ৪) Process  $P_2$  এর জন্য:

Work = Current Available = 532

Need ( $P_2$ ) = 600

সিরিজের দ্বারা নিয়ন্ত্রিত Preempted হয়ে থাকে যেমন যাদের মধ্যে একটি Process C।

Need ( $P_2$ )  $\leq$  Work ( $P_2$ )  
 $600 \leq 532$  is False  
So Finish ( $P_2$ ) = False.

If Available = [ 0 0 X 1 1 ], what is the smallest value of x for which this is a safe state?  
সমাধান:

প্রথমে আমাদের প্রতিটা প্রসেসের জন্য প্রত্যেক রিসোর্স type কত অতিরিক্ত instance লাগবে তা ক্যালকুলেট করতে হবে। তার জন্য Need ম্যাট্রিক্স নির্ণয় করতে হবে।

We know,  
Need = Maximum – Allocation

	Need				
A	0	1	0	0	2
B	0	2	1	0	0
C	1	0	3	0	0
D	0	0	1	1	0

Case-01: যদি X এর ভ্যালু 0 ধরি, তাহলে Available = [ 0 0 0 0 1 ]

উপরের আ্যাভাইলেবল ম্যাট্রিক্স দিয়ে কোন প্রসেস satisfy হবেনা, কেননা প্রতিটা প্রসেস, আ্যাভাইলেবলের চেয়ে বেশি লাগবে(need), যা নিত ম্যাট্রিক্স দেখে বুঝতে হবে: তাই বলা হয় X = 0 হলে ডেভলক থেকেই যাবে।

Case-02: যদি X এর ভ্যালু 1 ধরি, তাহলে Available = [ 0 0 1 1 ]

তথ্যাত্মক প্রসেস D satisfied হবে। প্রসেস D, requested resource কে allocate করবে। তার এক্স্রিবিউশন শেষ করে তার জন্য বরাদ্দ মেমরী খালী করবে।

তারপর, Available = [ 0 0 1 1 ] + [ 1 1 1 1 0 ]  
= [ 1 1 2 2 1 ]

উপরের আ্যাভাইলেবল ম্যাট্রিক্স দিয়ে আর কোন প্রসেস satisfy হবেনা, কেননা প্রতিটা প্রসেস, আ্যাভাইলেবলের চেয়ে বেশি লাগবে(need)। তাই বলা হয় X = 1 হলেও ডেভলক থেকেই যাবে।

Case-03: যদি X এর ভ্যালু 2 ধরি, তাহলে Available = [ 0 0 2 1 1 ]

তথ্যাত্মক প্রসেস D satisfied হবে। প্রসেস D, requested resource কে allocate করবে। তার এক্স্রিবিউশন শেষ করে তার জন্য বরাদ্দ মেমরী খালী করবে।

তারপর, Available = [ 0 0 2 1 1 ] + [ 1 1 1 1 0 ]  
= [ 1 1 3 2 1 ]

এরপর উপরের ম্যাট্রিক্স দিয়ে প্রসেস C satisfied হবে। প্রসেস C, requested resource কে allocate করবে। তার এক্স্রিবিউশন শেষ করে তার জন্য বরাদ্দ মেমরী খালী করবে।

তারপর, Available = [ 1 1 3 2 1 ] + [ 1 1 0 1 1 ]  
= [ 2 2 3 3 2 ]

এরপর উপরের আ্যাভাইলেবল ম্যাট্রিক্স দিয়ে প্রসেস A, B উভয়ই satisfied হবে। প্রসেস A, B requested resource কে

|
|  |

allocate করবে। তার এক্সিকিউশন শেষ করে তার জন্য বরাদ্দ মেমরী খালী করবে।

$$\text{তারপর, Available} = [22332] + [10211] + [20110] = [52653]$$

সবজেন্স ধারাবাহিক execute হয়েছে। তাই বলা যায় system is in a safe state। X এর ভালু মিনিমাম 2 হলে, সিস্টেম safe sequence থাকবে।

প্রশ্ন-১৫. A single processor system has three resource types X, Y and Z, which are shared by three processes. There are 5 units of each resource type. Consider the following scenario, where the column Allocation denotes the number of units of each resource type allocated to each process, and the column Request denotes the number of units of each resource type requested by a process in order to complete execution. Which of these processes will finish LAST?

	Allocation			Request		
	X	Y	Z	X	Y	Z
P <sub>0</sub>	1	2	1	1	0	3
P <sub>1</sub>	2	0	1	0	1	2
P <sub>2</sub>	2	2	1	1	2	0

- A. P<sub>0</sub>
- B. P<sub>1</sub>
- C. P<sub>2</sub>
- D. None of the above since the system is in a deadlock.

সমাধান:

According to question-

$$\text{Total} = [X \ Y \ Z] = [5 \ 5 \ 5]$$

$$\text{Total Allocation} = [X \ Y \ Z] = [5 \ 4 \ 3]$$

$$\begin{aligned} \text{Now, Available} &= \text{Total} - \text{Total Allocation} \\ &= [5 \ 5 \ 5] - [5 \ 4 \ 3] \\ &= [0 \ 1 \ 2] \end{aligned}$$

তথ্যমাত্র প্রসেস P<sub>1</sub> satisfied হবে। প্রসেস P<sub>1</sub>, requested resource কে allocate করবে। তার এক্সিকিউশন শেষ করে তার জন্য বরাদ্দ মেমরী খালী করবে।

$$\text{Then, Available} = [0 \ 1 \ 2] + [2 \ 0 \ 1] = [2 \ 1 \ 3]$$

এরপর আ্যাভালেক্ষন ম্যাট্রিক্স দিয়ে প্রসেস P<sub>0</sub> satisfied হবে। প্রসেস P<sub>0</sub> requested resource কে allocate করবে। তার এক্সিকিউশন শেষ করে তার জন্য বরাদ্দ মেমরী খালী করবে।

$$\text{Then-Available} = [2 \ 1 \ 3] + [1 \ 2 \ 1] = [3 \ 3 \ 4]$$

এরপর উপরের আ্যাভালেক্ষন ম্যাট্রিক্স দিয়ে প্রসেস P<sub>2</sub> satisfied হবে। প্রসেস P<sub>2</sub> requested resource কে allocate করবে। তার এক্সিকিউশন শেষ করে তার জন্য বরাদ্দ মেমরী খালী করবে।

$$\text{Then-Available} = [3 \ 3 \ 4] + [2 \ 2 \ 1] = [5 \ 5 \ 5]$$

Thus,

- ✓ There exists a safe sequence P<sub>1</sub>, P<sub>0</sub>, P<sub>2</sub> in which all the processes can be executed.
- ✓ So, the system is in a safe state. Process P<sub>2</sub> will be executed at last.

Thus, Option (C) is correct.

প্রশ্ন-১৬. An operating system uses the banker's algorithm for deadlock avoidance when managing the allocation of three resource types X, Y and Z to three processes P<sub>0</sub>, P<sub>1</sub> and P<sub>2</sub>. The table given below presents the current system state. Here, the Allocation matrix shows the current number of resources of each type allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution.

	Allocation			Max		
	X	Y	Z	X	Y	Z
P <sub>0</sub>	0	0	1	8	4	3
P <sub>1</sub>	3	2	0	6	2	0
P <sub>2</sub>	2	1	1	3	3	3

There are 3 units of type X, 2 units of type Y and 2 units of type Z still available. The system is currently in Safe State. Consider the following independent requests for additional resources in the current state-

REQ1: P<sub>0</sub> requests 0 units of X, 0 units of Y and 2 units of Z

REQ2: P<sub>1</sub> requests 2 units of X, 0 units of Y and 0 units of Z

Which of the following is TRUE?

- A. Only REQ1 can be permitted.
- B. Only REQ2 can be permitted.
- C. Both REQ1 and REQ2 can be permitted.
- D. Neither REQ1 nor REQ2 can be permitted.

সমাধান:

According to question (প্রশ্নমতে),

$$\text{Available} = [X \ Y \ Z] = [3 \ 2 \ 2]$$

Now,

$$\text{Need} = \text{Max} - \text{Allocation}$$

So, we have-

	Allocation			Max			Need		
	X	Y	Z	X	Y	Z	X	Y	Z
P <sub>0</sub>	0	0	1	8	4	3	8	4	2
P <sub>1</sub>	3	2	0	6	2	0	3	0	0
P <sub>2</sub>	2	1	1	3	3	3	1	2	2

Currently, the system is in safe state.  
(It is given in question. If we want, we can check)

Checking Whether REQ1 Can Be Entertained-

$$\text{Need of } P_0 = [0 \ 0 \ 2]$$

$$\text{Available} = [3 \ 2 \ 2]$$

Clearly,

- ✓ With the instances available currently, the requirement of REQ1 can be satisfied.
- ✓ So, banker's algorithm assumes that the request REQ1 is entertained.
- ✓ It then modifies its data structures as-

	Allocation			Max			Need		
	X	Y	Z	X	Y	Z	X	Y	Z
P <sub>0</sub>	0	0	3	8	4	3	8	4	0
P <sub>1</sub>	3	2	0	6	2	0	3	0	0
P <sub>2</sub>	2	1	1	3	3	3	1	2	2

$$\begin{aligned} \text{Available} &= [3 \ 2 \ 2] - [0 \ 0 \ 2] \\ &= [3 \ 2 \ 0] \end{aligned}$$

- ✓ Now, it follows the safety algorithm to check whether this resulting state is a safe state or not.

- ✓ If it is a safe state, then REQ1 can be permitted otherwise not.

Step-01:

- ✓ With the instances available currently, only the requirement of the process P<sub>1</sub> can be satisfied.
- ✓ So, process P<sub>1</sub> is allocated the requested resources.
- ✓ It completes its execution and then free up the instances of resources held by it.

Then-

$$\text{Available} = [1 \ 2 \ 2] + [5 \ 2 \ 0] = [6 \ 4 \ 2]$$

Step-02:

- ✓ With the instances available currently, only the requirement of the process P<sub>2</sub> can be satisfied.
- ✓ So, process P<sub>2</sub> is allocated the requested resources.

$$= [6 \ 4 \ 0]$$

Now,

- ✓ It is not possible to entertain any process.
- ✓ The system has entered the deadlock state which is an unsafe state.
- ✓ Thus, REQ1 will not be permitted.

Checking Whether REQ2 Can Be Entertained-

$$\text{Need of } P_1 = [2 \ 0 \ 0]$$

$$\text{Available} = [3 \ 2 \ 2]$$

Clearly,

- ✓ With the instances available currently, the requirement of REQ1 can be satisfied.
- ✓ So, banker's algorithm assumes the request REQ2 is entertained.
- ✓ It then modifies its data structures as-

	Allocation			Max			Need		
	X	Y	Z	X	Y	Z	X	Y	Z
P <sub>0</sub>	0	0	1	8	4	3	8	4	2
P <sub>1</sub>	5	2	0	6	2	0	1	0	0
P <sub>2</sub>	2	1	1	3	3	3	1	2	2

$$\text{Available} = [3 \ 2 \ 2] - [2 \ 0 \ 0]$$

$$= [1 \ 2 \ 2]$$

- ✓ Now, it follows the safety algorithm to check whether this resulting state is a safe state or not.
- ✓ If it is a safe state, then REQ2 can be permitted otherwise not.

- ✓ It completes its execution and then free up the instances of resources held by it.

Then-

$$\text{Available} = [642] + [211] = [853]$$

**Step-03:**

- ✓ With the instances available currently, the requirement of the process  $P_0$  can be satisfied.
- ✓ So, process  $P_0$  is allocated the requested resources.
- ✓ It completes its execution and then free up the instances of resources held by it.

Then-

$$\begin{aligned}\text{Available} &= [853] + [001] \\ &= [854]\end{aligned}$$

Thus,

- ✓ There exists a safe sequence  $P_1, P_2, P_0$  in which all the processes can be executed.
- ✓ So, the system is in a safe state.
- ✓ Thus, REQ2 can be permitted.

Thus, Correct Option is (B).

**Ques-17.** নিম্নে একটি সিস্টেম দেওয়া হলো, যা কিনা ডেডলক এভিয়ে  
(Deadlock Avoid) চলার জন্য ব্যাকার্স এলগরিদম ব্যবহার করে।

Process	Allocation	Max	Available
$P_0$	0012	0012	1520
$P_1$	1000	1750	
$P_2$	1354	2356	
$P_3$	0632	0652	
$P_4$	0014	0656	

ব্যাকার্স এলগরিদম ব্যবহার করে নিচের প্রশ্ন তলো সমাধান করুন:

- (ক) Need Matrix কী হবে?  
(খ) উল্লিখিত সিস্টেম কী Safe State এ যাবে?  
(গ)  $P_1$  Process থেকে যদি  $(0, 4, 2, 0)$  একটি Request আসে, তা কি সাথে সাথে Granted হবে?
- উত্তর: Try Yourself. (নিজে চেষ্টা করুন)

### Memory Management Techniques

(Paging, Segmentation and page Replacement Policies)

**Ques-18.** Memory Management বলতে কি বুঝায়? এর প্রয়োজনীয়তা লিখুন।

উত্তর: Memory Management: Operating System কর্তৃক User এর জন্য মেমোরির ব্যবস্থাপনা করে। এটি করাকে Memory Management বলে।

প্রয়োজনীয়তা: Multiprogramming System এ বিভিন্ন প্রসেস সমূহের মধ্যে মেমোরি ম্যানেজমেন্ট এর মাধ্যমে সমস্যা সাধন করে। সর্বাঙ্গেক অনুচ্ছেদে অবস্থা স্থিতি করা সহজ। এতে সিস্টেম এখন বৃক্ষ পায়, মেমোরির অপচয় ঘোষ হয়।

**Ques-19.** Virtual Memory বলতে কী বোঝায়? এর কার্যপদ্ধতি সংক্ষেপে বর্ণনা করুন। [NTRCA (Lecturer)-2019] [Elections Commission-12]

উত্তর: **Virtual Memory:** Virtual Memory হল Secondary Memory (সহায়ক স্মৃতি) এর এমন একটি অবস্থা যেখানে কম্পিউটারের Secondary Memory (সহায়ক স্মৃতি) এর একটি অংশ বিশেষকে Main memory হিসেবে ব্যবহার করে। ফলে প্রথম মেমোরির তথ্য ধারান ক্ষমতার সীমাবদ্ধতা দূর করার জন্যই এ করা অবশ্যিনী।

**Virtual Memory** এর কার্যপদ্ধতি: ধরি একটি অপারেটিং সিস্টেমে 120MB দরকার চলমান সব প্রয়োজন ধরে রাখার জন্য। কিন্তু সেক্ষেত্রে র্যাম চিপে ফিজিক্যাল মেমোরি হিসেবে মাত্র 50MB ব্যবহার যায়। আছে। তখন অপারেটিং সিস্টেম Virtual Memory হিসেবে 120MB সেট আপ করে যা কিনা Virtual Memory Manager (VMM) ব্যবহার করবে। এই VMM হার্ডডিকের মধ্যে  $(120-50) = 70\text{MB}$  সাইজের। অর্থাৎ যতটুকু প্রয়োজন ততটুকু মধ্যে একটি ফাইল তৈরি করবে। যাকে পেজ-ফাইল বলে। আর তার অপারেটিং সিস্টেম মেমোরিকে এন্ড্রেস করতে পারবে এবং সত্যিকার অর্থে RAM এর মধ্যে 120MB এর ডাটা সংরক্ষণ করতে পারবে। এই সম্পূর্ণ কাজটা Virtual Memory Manager (VMM) ই করে।

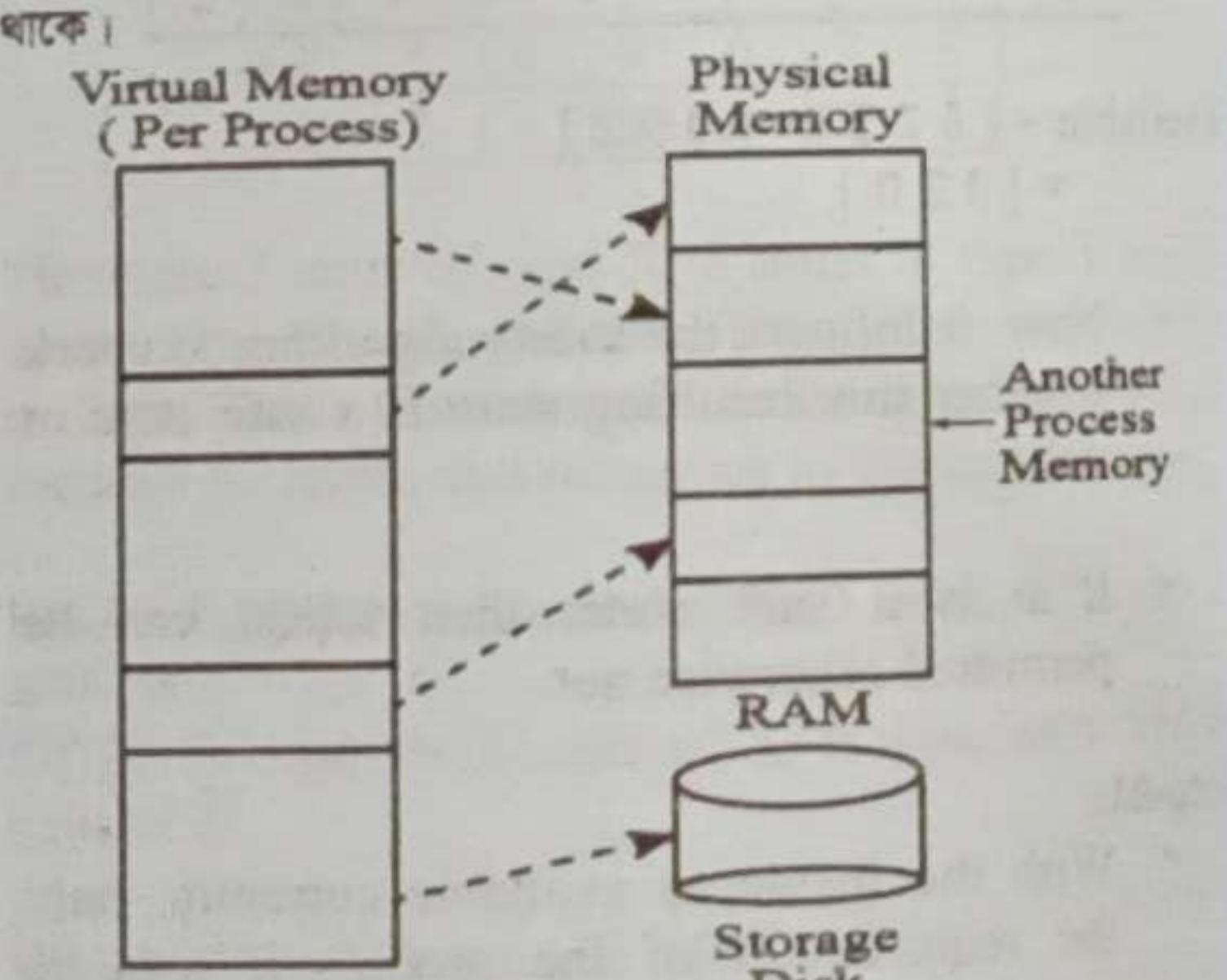


Figure: Virtual Memory এর কার্যপদ্ধতি

**Ques-20.** Virtual Memory এর প্রয়োজনীয়তা ব্যাখ্যা করুন।  
বালিজ এক্সামেন- ২০২১-সহকারী প্রয়োজন।

উত্তর: কোন ফাইল ওপেন করলে তা র্যামে লোড হয়। যদি ফাইলের সাইজ বেশি হয় অর্থাৎ বেশি পরিমাণ ডেটা নিয়ে কাজ করা হলে সমস্যা সৃষ্টি হয়। একেকে উইঙ্গেজ অপারেটিং সিস্টেমে সেকেভারি মেমোরি হার্ডডিকের কিছু স্পেসকে ফিজিক্যাল মেমোরি হিসাবে ব্যবহার করা যায়। একে ভার্চুয়াল মেমোরি বলা হয়।

নিচে বিভিন্ন প্রয়োজনীয়তা দেখানো হল:

- ❖ **memory যাটি**
  - দক্ষ যেকোন পরিচালনার প্রয়োজন।
  - physical memory জন্য প্রক্রিয়াটি খুব বড় হলে।
  - physical memory দেখে বেশি সক্রিয় প্রক্রিয়া ধরে রাখতে হলে।
- ❖ **একাধিক প্রয়োজনীয়তার প্রয়োজনীয়তা**
  - দক্ষ সুরক্ষা পক্ষতি
  - সহজে ভাগ করে নেয়া যায়।

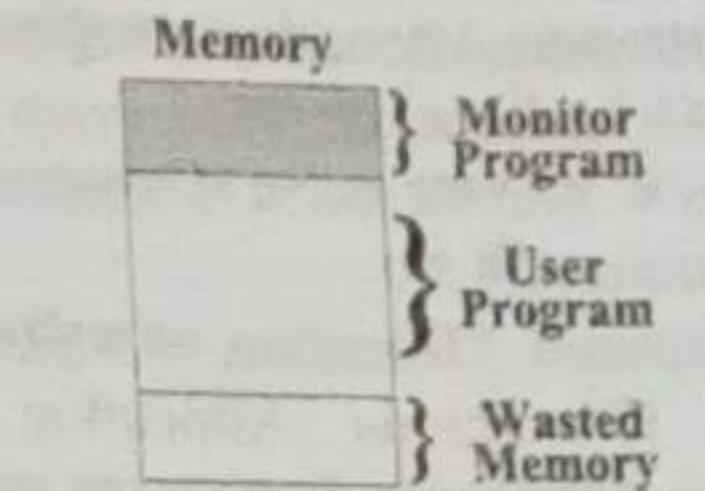


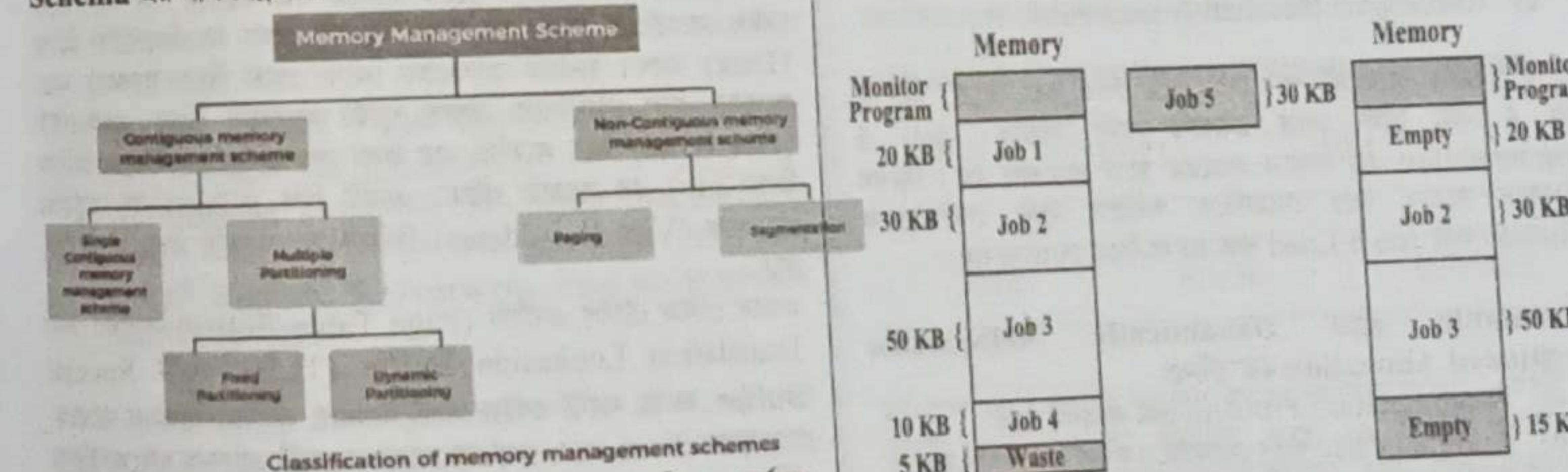
Figure: Single Contiguous Allocation

**Single Contiguous Allocation Memory Management** এর সুবিধা ও অসুবিধা সমূহ

- a) এটি একটি সহজ পক্ষতি।  
b) অটিলতা কম।  
c) কমদার্মা।  
d) অতিরিক্ত হার্ডওয়্যার Facility দরকার হয় না।  
অসুবিধা:  
a) প্রয়োজনের সাইজ ছোট হলে মেমোরির বিরাট একটা অংশ অব্যবহৃত অবস্থায় থাকে।  
b) সিপিইউ অধিক সময় Idle থাকে।

**(ii). Multiple Partitioned Allocation Memory Management** পক্ষতির বর্ণনা: Partition Allocation System পক্ষতিতে একাধিক জব প্রতিটির জন্য স্মৃতিতে জায়গা ব্যাপক থাকে। প্রতিটির জন্য টাইম শ্লাইস এবং Priority এর ভিত্তিতে সিপিইউ জব সমূহকে নির্ধারিত করতে থাকে। নির্ধারিত কাজ বাস্তু রাখে যান।

- a) জবের নির্ধারিত কাজ শেষ হয়।  
b) Error দেখা দিলে।  
c) I/O অপারেশনের প্রয়োজন হলে।  
d) জবের জন্য নির্ধারিত সময় শেষ হয়ে গেলে।
- টাইম শ্লাইস (Time Slice) ভিত্তিতে জব সম্পর্ক করতে থাকে। এভাবে একটি জব সম্পর্ক হওয়ার পর জবের জন্য স্মৃতির নির্ধারিত অংশ থালি হয়ে যায়। এই থালি অংশ তার ছোট মেমোরি বা সমান মেমোরির খণ্ডের কোন জব দ্বারা পূরণ হতে পারে।



**(i). Single Contiguous Allocation** পক্ষতির বর্ণনা: Single Contiguous Allocation Memory Management এর সবচেয়ে সরল পক্ষতি। এই পক্ষতিতে Memory তে তখন মাত্র একটি Program Execute হয়।

- a) Primary Memory.  
b) Job1, Job4 সম্পর্ক হওয়ার Management পর Memory এর অবস্থা।

- Multiple Partitioned Allocation** এর সুবিধা:
- CPU এর Idle Time হ্রাস পায়।
  - Memory'র অব্যবহৃত অংশের পরিমান হ্রাস পায়।
  - Throughput বৃদ্ধি পায়।

#### Multiple Partitioned Allocation এর অসুবিধা:

- Single Contiguous Allocation এর তুলনায় Partition Allocation System বায় বজ্জ্বল।
- একাধিক জব একই সাথে মেইন মেমোরিতে অবস্থান করে বলে একটি জবের সাথে অন্য একটি জবের যাতে ছিন্ন না ঘটে সেই জন্য নিরাপত্তা ব্যবস্থা প্রয়োজন হয়।
- বৃহৎ মেমোরির প্রয়োজন।
- Fragmentation সমস্যা দেখা যায়।

(iii). **Relocatable and Dynamically Relocatable Partitioned Allocation** পদ্ধতি এই Schema তে কোন Fragmentation সমস্যা থাকে না বা ইহা Fragmentation Reduce করার একটি উপায়।

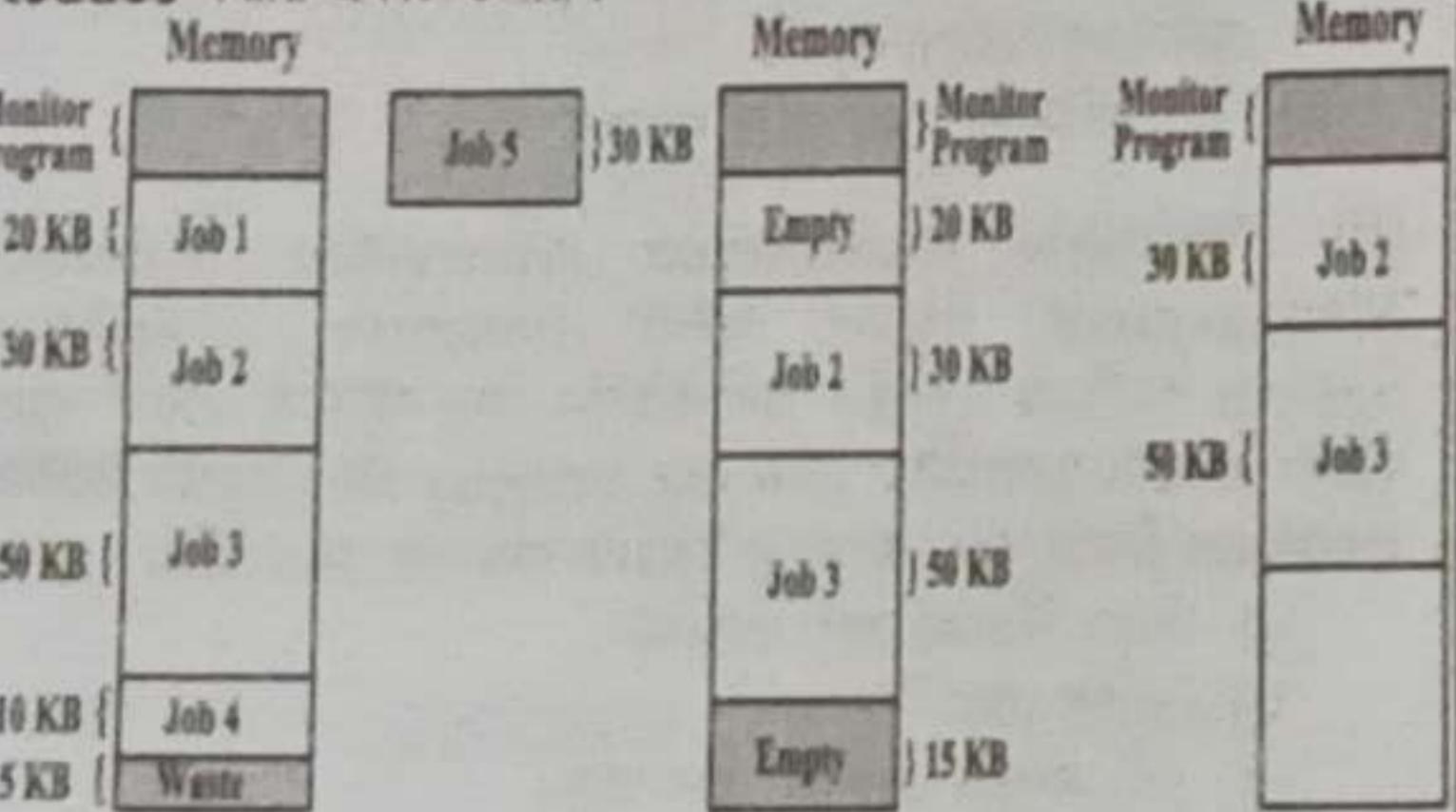


Figure: Relocatable and Dynamically Partitioned Allocation

- Initial
- Job1 & Job4
- Recompact Finished & load Job5

এখানে প্রাইমারি ভাবে ৫টি জব মেমোরিতে লোড করা আছে। অন্তর্পর Job 4 এর কাজ শেষে মেমোরি খালি হয়েছে। Job 5 Fragmentation এর কারণে লোডেড হতে বাধাপ্রাপ্ত হয়। বিশেষ হার্ডওয়্যার প্রয়োগ করে মেমোরিতে অবস্থিত বাকী সেলস্টোকে Relocate করে Job 5 Load করা হয় যা ঠিকে সেবানো হল।

#### Relocatable and Dynamically Relocatable Partitioned Allocation এর সুবিধা:

- Fragmentation Problem নেই বললেই চলে।
- Automatically ভাবে মেমোরিতে নতুন Job লোড হয়।
- CPU এর Idle time হ্রাস পায়।
- মেমোরি এর সর্বোত্তম ব্যবহার হয়।
- Job এর এড্রেস সম্পূর্ণ বাদীন থাকে।

#### Relocatable and Dynamically Relocatable Partitioned Allocation এর অসুবিধা:

- অতিরিক্ত হার্ডওয়্যার এর প্রয়োজন।

- ইহা একটি ব্যাবহৃত পদ্ধতি।
- কিছু Core মেমোরি নষ্ট হয়ে থাকে।
- Relocation Cost অনেক বেশি।
- Partition Allocation এর তুলনায় CPU time কম।

#### প্রক্ষ-৭. Paged Allocation ও Segmented Page allocation বর্ণনা কর।

**উভয় Paged Allocation:** যে পদ্ধতিতে Job address unit কে page আকারে এবং Physical memory কে কতগুলো frame আকারে ভাগ করে নিয়ে কার্য পরিচালনা করা হয় তাকে Paged Allocation বলে।

**Segmented Paged allocation:** যে পদ্ধতিতে Segmented & Paged Memory Management System ব্যবহার করে Memory Allocation করা হয় তাকে Segmented Paged allocation বলে।

#### প্রক্ষ-৮. Segmentation এর সুবিধা লিখ।

##### উভয় সুবিধা:

- থেকেকটি Segment প্রবেশের অধিকার সংরক্ষনের জন্য বিশেষ ধরনের Attribute ব্যবহার করা।
- যদি ALPHA সেগমেন্ট নির্বাচন হয়, তবে Hardware দ্বারা Instruction কে অনুমতি প্রদান করে না।

প্রক্ষ-৯. Demand Paging (ডিম্যান্ড পেজিং) বর্ণনা কর। বা, চিন্তাহ ভার্চুয়াল মেমোরি পদ্ধতি এর Demand Paging (ডিম্যান্ড পেজিং) বর্ণনা কর।

**উভয় ভার্চুয়াল মেমোরির Simple Idea** (সহজ ধারণা) হচ্ছে ব্যাক আকারের প্রোগ্রাম, ডেটা ও Stack (স্ট্যাক), যা মেশিনের নির্ধারিত মেমোরি সাইজের চেয়ে বেশি। তাকে এই নির্ধারিত মেমোরিতে Execute (নির্বাচন) করার ব্যবস্থা করা। অপারেটিং সিস্টেম এ পদ্ধতিতে প্রোগ্রামের যে অংশটি নিন্দিত মূলতে ব্যবহার করা দরকার, সে অংশটির মেইন মেমোরিতে রাখে এবং প্রোগ্রাম ও অন্যান্য অংশগুলোকে ডিস্ক (Disk) রাখে। অবশ্যিক অংশগুলো থেকে কোন অংশ ব্যবহার করা প্রয়োজন হলে মেমোরিতে রাখিক্ত পূর্বের অংশটির উপর ওভারলেট (Overwrite) করে প্রসেসিং এর কাজ সম্পন্ন করে। ইহা একাধিক চিপ (Chip) এর সময়ে গঠিত। একটি চিপ (Chip) বা ভার্চুয়াল এড্রেসকে (Virtual Address) ফিজিক্যাল মেমোরির এড্রেসে মাঝি করে।

এখানে পেজ টেবিল এনটিটি (Page Table Entity) সম্বরে জন Translation Lookaside Buffer (TLB) একটি Special Buffer, যা যে সমস্ত পেজ এনটিটি তলো এইমাত্র ব্যবহার হয়ে, সেগুলোকে Store করে। ভার্চুয়াল এড্রেস অনুযায়ী প্রসেসর প্রথমে TLB পরীক্ষা করে। যদি প্রয়োজনীয় পেজ টেবিল এনটিটি পাওয়া যায় (A TLB Hit), তবে Frame Number Retrieve হয়ে Real Address নির্ধারিত হয়।

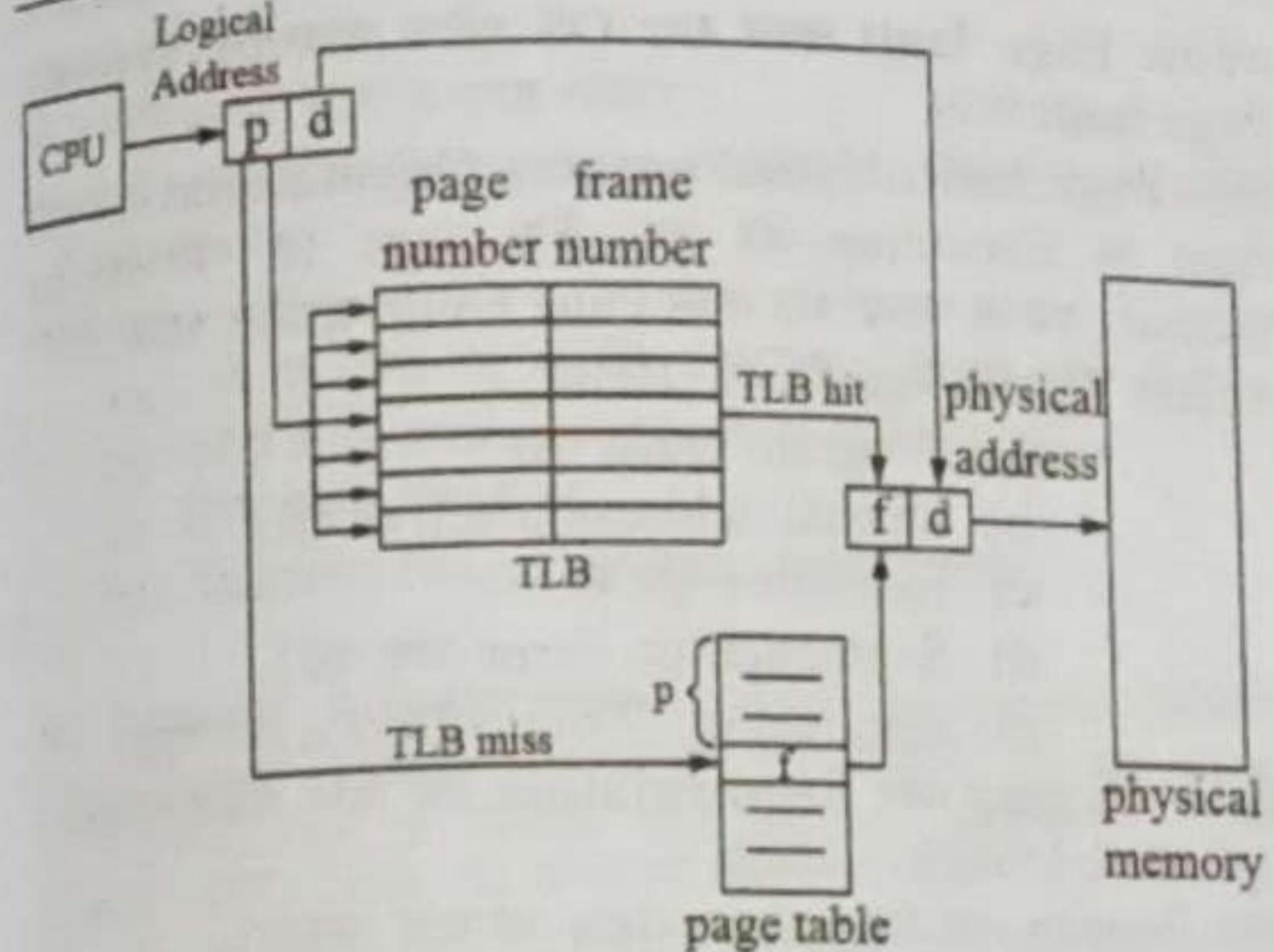


Figure: Translation Lookaside Buffer (TLB)

**পেজ টেবিল:** পেজ টেবিলটি এমন একটি তেটা স্টোকচার যা ভার্চুয়াল মেমোরি সিস্টেম ব্যবহার করে ফিজিক্যাল অ্যাড্রেস এবং ফিজিক্যাল অ্যাড্রেস এর মধ্যে ম্যাপিং করে।

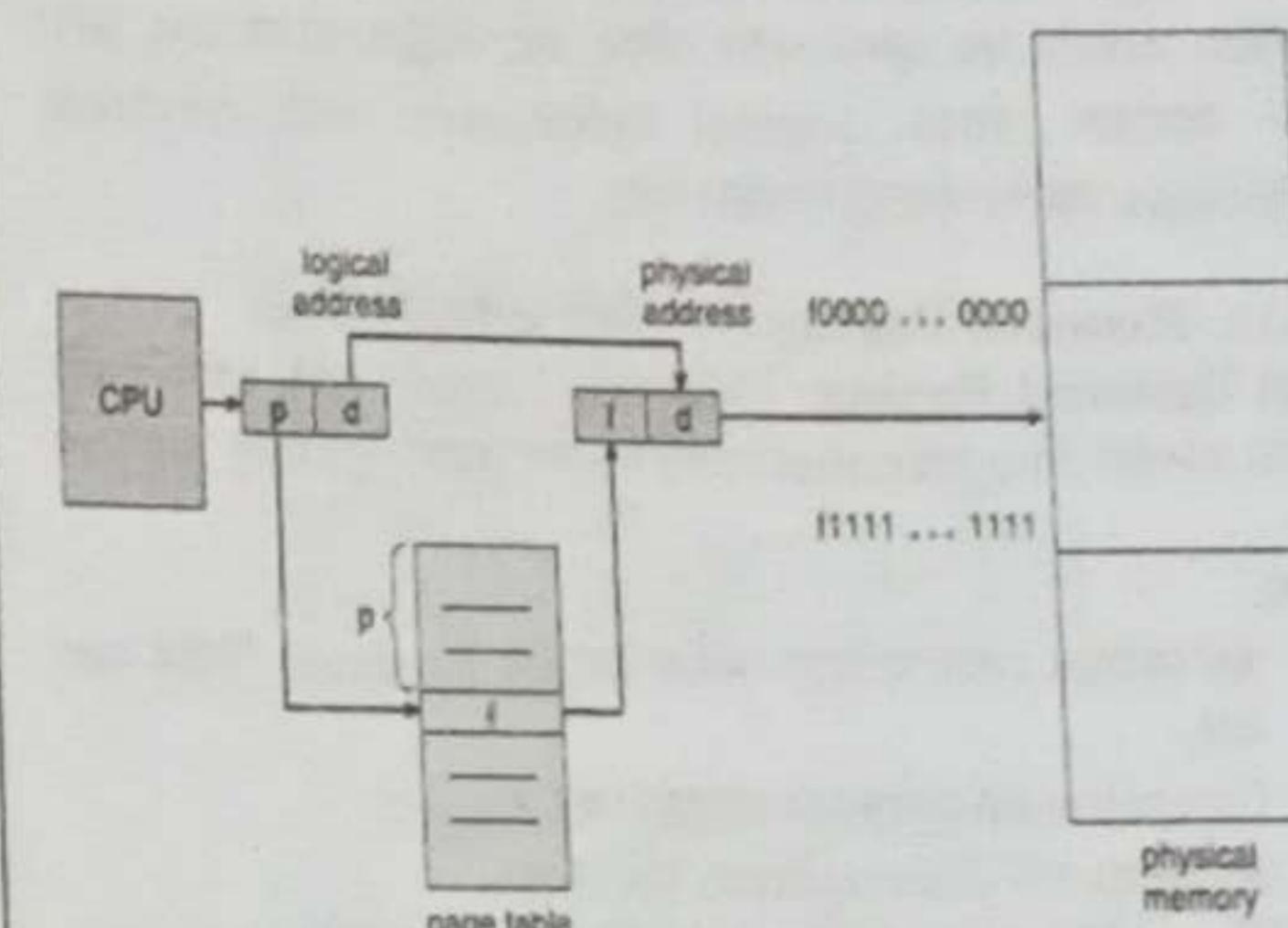


Fig: Page Table

**ফিজিক্যাল অ্যাড্রেস:** পেজ টেবিলের প্রসেসের জন্য সিপিইউ দ্বারা ফিজিক্যাল অ্যাড্রেস তৈরি করা হয় যা সাধারণত প্রসেসিং এর জন্য ব্যবহৃত হয়।

**ফিজিক্যাল অ্যাড্রেস:** ফিজিক্যাল অ্যাড্রেস হলো মেমোরি একচুলাল ফ্রেম অ্যাড্রেস যা সাধারণত হার্ডওয়্যার বা বিশেষ ব্যাম সাবসিস্টেম গুলিতে ব্যবহৃত হয়।

সিপিইউ ফিজিক্যাল অ্যাড্রেসকে পেজ নম্বর এবং পেজ অফসেট বিভক্ত করে। উপরের ছবিতে মেমোরি ইউনিট নামে একটি ইউনিট দেখা যাচ্ছে। যা ফিজিক্যাল অ্যাড্রেসের পেজ নম্বরটিকে ফিজিক্যাল অ্যাড্রেসের ফ্রেম নম্বরে জৰাজৰ করে। উভয়ের অফসেট অ্যাড্রেস একই থাকে।

মেমোরি ম্যানেজমেন্ট ইউনিটের জন্য বিশেষ ধরণের ম্যাপিং দরকার যা পেজ টেবিল দ্বারা করা হয়। পেজ টেবিলের পেজ নম্বরটিকে সম্পর্কিত সমষ্ট ফ্রেম নম্বর সংরক্ষণ করে থাকে।

একটি 32-bit সিস্টেমের উদাহরণ: লিনার অ্যাড্রেসের ফর্ম্যাট

31	22 21	12 11	0
+	- - +	- - +	-
DIR   PAGE   OFFSET			
+	- + -	- + -	- + -

আমাদের কাছে 32-bit ভার্চুয়াল অ্যাড্রেসের দৈর্ঘ্য রয়েছে। প্রতিটি পেজ ফ্রেমের জন্য অফসেট অ্যাড্রেস হলো 12 বিটের একটি পেজ ডিভেলি আর প্রতিটি পেজ ফ্রেমের অ্যাড্রেস হলো একটি পেজ টেবিল ডিভেলি আর প্রতিটি পেজ টেবিলে, পেজ ফ্রেমের অ্যাড্রেসটি পছন্দসই অ্যাড্রেস। বিতীয় স্তরের পেজ টেবিলে, পেজ ফ্রেমের অ্যাড্রেসটি পছন্দসই অ্যাড্রেস। মেমোরি অপারেট ধারণ করে থাকে। 1024 পেজ রয়েছে যা বিতীয় স্তরের পেজ অফসেটের জন্য 10 ৰিট লাগবে। এবং 1024 পেজ টেবিল রয়েছে যা পেজ ডিভেলির জন্য 10 ৰিট লাগবে। এটিই 32-bit অ্যাড্রেস ফর্ম।

**পেজ ফাল্ট (Page Fault):** যখন অপারেটিং সিস্টেমে ভার্চুয়াল মেমোরিতে তেটা খুঁজে না পায় তখন পেজ ফাল্ট ঘটে।

**প্রশ্ন-১১. Operating System এ Page Table ব্যবহার করা হয়**  
কেন? [৩০ মোদারেগ ও অক্ষিত বিজ্ঞান-২০১৯-তাঁটা এন্ট্রি সুপারভাইজার]  
উত্তর: Page Table হল একটি ডাটা স্ট্রাকচার যা ভার্ফল মেমরি সিস্টেমে ব্যবহার হয়। মূলত ডাটা স্টের এবং রিট্রিভ কালে এবং ডাটা দ্রুত আস্রেস করতে logical addresses এবং physical addresses মাপিং করতে ব্যবহৃত হয়।

**প্রশ্ন-১২. Demand Paging কি? সুবিধা ও অসুবিধা লিখ।**  
উত্তর: Demand Paging: Demand Paging হলো এমন একটি পদ্ধতি যেখানে Programmer দের বপ্প বা কলন পুরোপুরি ব্যবহারণ হয়।

**সুবিধা:**

- ব্যবহৃত কোর স্থূলতে অধিক সৈর্বের Program নির্বাচন করা যায়।
- Operator এর মধ্যে প্রয়োজন হয় না।
- System এর Throughput বৃদ্ধি করে।
- Time Sharing পদ্ধতির জন্য এটি বেশি উপযোগী।
- Programmer Core Memory কে Large Memory হিসেবে কলন করতে পারে এবং নিজের প্রয়োজনকে Multiprogramming পরিবেশে নির্বাচন করতে পারে।

**অসুবিধা:**

- যখন অধিক সংখ্যক Program Time Share করে ব্যবহৃত Core Memory কে ব্যবহার করে, তখন Throughput করে যায়।
- Paging Judgement Algorithm অপেক্ষাকৃত নিম্নমানের পেজকে Core স্থূলতে নিয়ে পড়া, ছানাস্বর করা এবং পুনরায় সহায়ক Storage এ Load করার ক্ষেত্রে System এর Throughput অত্যাক্ত করে যায়।

**প্রশ্ন-১৩. তিনিমিত Segmented Allocation বর্ণনা করুন।**  
উত্তর: Segmented Allocation: Segment হচ্ছে একপ্রকার তথ্যের সমষ্টি, যা একটি এন্টিটির ন্যায় আচরণ করে। আবার ইয়া এক বা একাধিক প্রোগ্রাম, একটি ডেটাবেজ বা একাধিক ডেটাবেজ ও হতে পারে। সেগমেন্টেড অবস্থায় এক্সেস সমূহ বিমানিক (Two Dimensional) হয় এবং এক্সেস দুটি কম্পোনেন্ট (Component) দ্বারা প্রকাশ করা হয়। যেমন:

- Name of Segment (সেগমেন্টের নাম)।
- Name of Offset (অফসেটের নাম)।

এতে প্রত্যেকটি প্রসেসের নিজের সেগমেন্ট টেবিল থাকে, যখন কোন প্রসেস এর সকল সেগমেন্ট মেইন মেমোরিতে লোড হয়, তখন প্রসেসের জন্য সেগমেন্ট টেবিল তৈরি হয় এবং মেইন মেমোরিতে লোড হয়। সব সেগমেন্ট টেবিলের এন্টিটি, মেইন মেমোরিতে কোন নির্দিষ্ট সেগমেন্টের এবং সেগমেন্টের Length ধারণ করে। কোনো Word কে মেমোরি থেকে Read করার ক্ষেত্রে ভার্ফল বা লজিক্যাল এক্সেস অনুসূব্দ করে সেগমেন্ট টেবিলের সাহায্যে ফিজিক্যাল এক্সেস তৈরি করে। যেহেতু সেগমেন্ট টেবিল Variable Length এর হয়, সে জন্য একে এক্সেস করার জন্য মেইন মেমোরিতে নিতে হয়। প্রত্যেক সেগমেন্টেই নিজের Access অধিকার সংশ্লিষ্ট থাকে এবং Segmentation, Dynamic Linking সুবিধা প্রদান করে। এইজন্য একে Dynamic Linking ও বলা হয়।

**প্রশ্ন-১৪. Page fault কখন হয়? OS গৃহীত কাজগুলো লিখ যখন Page fault ঘটে?**  
উত্তর: Page fault: Virtual memory Organization এ কোন Page কে Execution এর সময় উক্ত page কে যদি Main Memory তে না পাওয়া যায় তখন Page Fault সংগঠিত হয়। Action When Page Fault Occur:

- Program crash করে।
- Virtual Address Inactive হয়।
- Bus Error সৃষ্টি হয়।
- Segmentation Error দেখা হয়।
- Application টি Shut Down হয়ে যায়।

**প্রশ্ন-১৫. Paging এবং Segmentation এর মধ্যে পার্থক্য লিখ।**  
উত্তর: Paging এবং Segmentation এর মধ্যে পার্থক্য:

Paging	Segmentation
Paging নির্দিষ্ট সাইজের ত্বক আকারের হয়।	Segmentation যে কোন সাইজের হয়ে থাকে।
পেজ এর Internal Fragmentation এর ক্ষেত্রে হয়।	Segmentation External Fragmentation এর ক্ষেত্রে হয়।
হার্ডওয়ার Paging এর সাইজ User নির্ধারণ করে।	Segmentation এর সাইজ User নির্ধারণ করে।
Page Table প্রতিটি Paging এর Base Address ধারণ করে।	Segmentation Table প্রতিটি Segment এর Segment Number এবং Offset Number ধারণ করে।

**প্রশ্ন-১৬. বিভিন্ন ধরনের ফাইল সিস্টেম তালুর নাম ও তাদের সুবিধা অসুবিধা তালু লিখুন।**

উত্তর: বিভিন্ন ধরনের ফাইল সিস্টেম নিম্নরূপঃ

- Contiguous Allocation: একক সময়ে একটি Program Disk থেকে Load হয়। Disk Allocation System এর সবচেয়ে সরল ও সহজ পদ্ধতি হচ্ছে Contiguous Allocation System। এই পদ্ধতিকে Disk Address ব্যবহার করে Linear Ordering এ তিক তালিকে Identify করা যায়। উহুর জন্য একটি Directory Table থাকে। n সংখ্যক ডিক স্পেস থাকলে এদের B, B+1, B+2, ..., B+N-1 ইত্যাদি দ্বারা Addressing করা থাকে। প্রতিটি ফাইল এর Directory Entity এর মাধ্যমে ফাইল এর Starting Block Address ও File এর জন্য বন্টনকৃত এরিয়া কে বুকানো হয়।

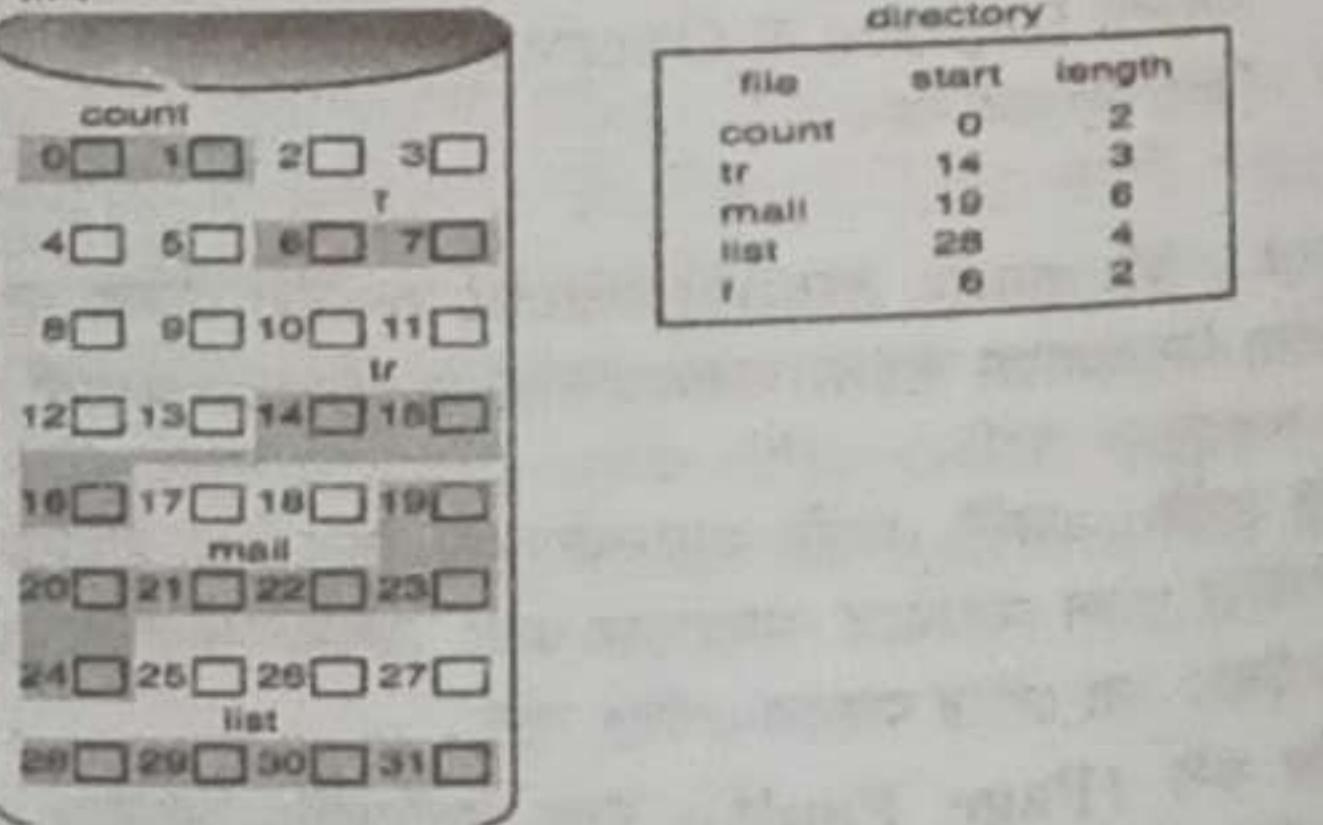


Figure: Contiguous Allocation

**সুবিধা:**

- এটি একটি সরল ও সহজ পদ্ধতি।
- এই পদ্ধতিতে Sequential ও Direct উভয় Accessing সহজ।

**অসুবিধা:**

- এই পদ্ধতিতে নতুন ফাইল এর স্পেস খুজে বের করা কঠিন।
- এতে External Fragmentation সমস্যা ঘটে।
- তিক স্পেস অপচয় হয়।
- Internal Fragmentation ও ঘটতে পারে।

b) **Linked Allocations:** Linked Allocation পদ্ধতিতে Contiguous Allocation পদ্ধতির সকল অসুবিধা দূর করা হয়েছে। এ পদ্ধতিতে File এ Block সমূহ Linked অবস্থায় থাকে। তাই Block তালু Disk এর বেছানেই থাকুক না কেন File এর First Block এর উক্ততে ও শেষে Block এ একটি Pointer থাকে। তিনে দেখা যায় যে, 5 টি Block এর সময়ে ঘটিত একটি File এর Starting Block হচ্ছে 2. Directory থেকে তাই Block 2 এর জন্য একটি Pointer এর ব্যবহার করা আছে যা প্রবর্তী Block তালুর সাথে (জমারে 13,18,12) Linked হয়ে সর্বশেষ Block 7 এ শেষ হয়েছে। আবার সর্বশেষ Block হিসাবে Directory থেকে এর জন্য পূর্বের একটি Pointer এর ব্যবহার করা হয়েছে।

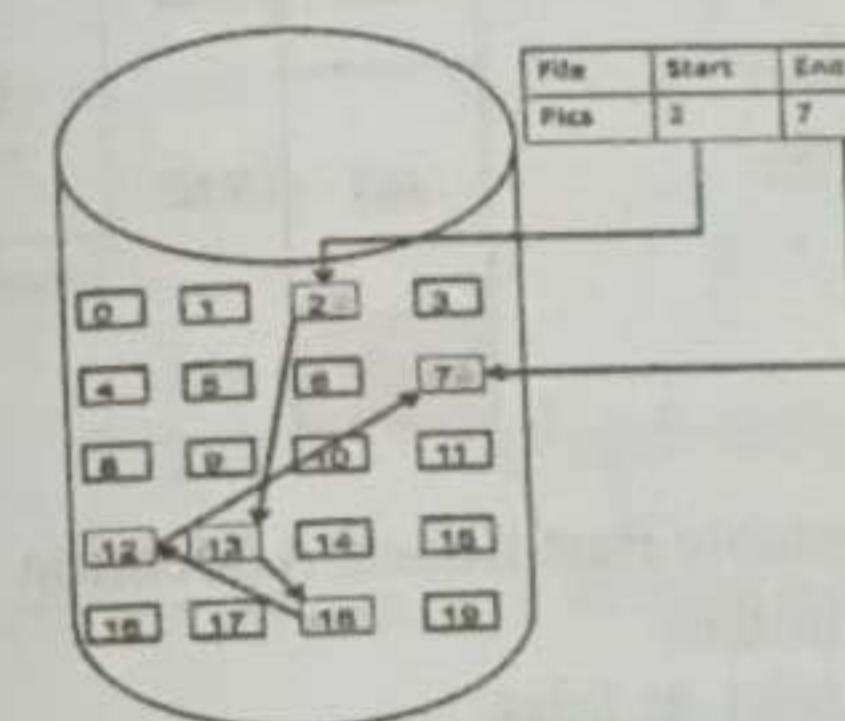


Figure: Linked Allocation

**সুবিধা:**

- এই পদ্ধতিতে Fragmentation ছাড়াই Direct Access পদ্ধতি অনুসূব্দ করে।
- ফাইল তৈরির সময় ফাইল সাইজ তৈরি করতে হয় না। ফাইল সাইজ নির্দিষ্ট করার প্রয়োজন হয় না।
- কোন External Fragmentation ঘটেনা।

**অসুবিধা:**

- Direct File Access করা জটিল।
- Pointer অতিরিক্ত Space দখল করে।
- এটি যথেষ্ট Reliable নয়।
- প্রত্যেকটি Block একটি Pointer দখল করে, ফলে মেমোরির অপচয় হয়।
- Blocks স্বত্বান্তে ছড়িয়ে ছিটয়ে থাকে।

(iii). **Indexed Allocation:** Pointer জনিত সমস্যা সমাধানে এ পদ্ধতিতে সকল Pointer সমূহকে একটি মাত্র নির্দিষ্ট Location এ সকল Pointer সমূহকে একত্রিত করা হয়েছে এবং এর নাম দেওয়া হয়েছে Index Block. Index Block এ প্রতিটি File এরই

Index Block রয়েছে যাকে Disk Block Address এর Array হিসাবে বিবেচনা করা যায়। Index Block এর n সংখ্যক Entry কে File এর n সংখ্যক Block হিসাবেই Point করা যায়।

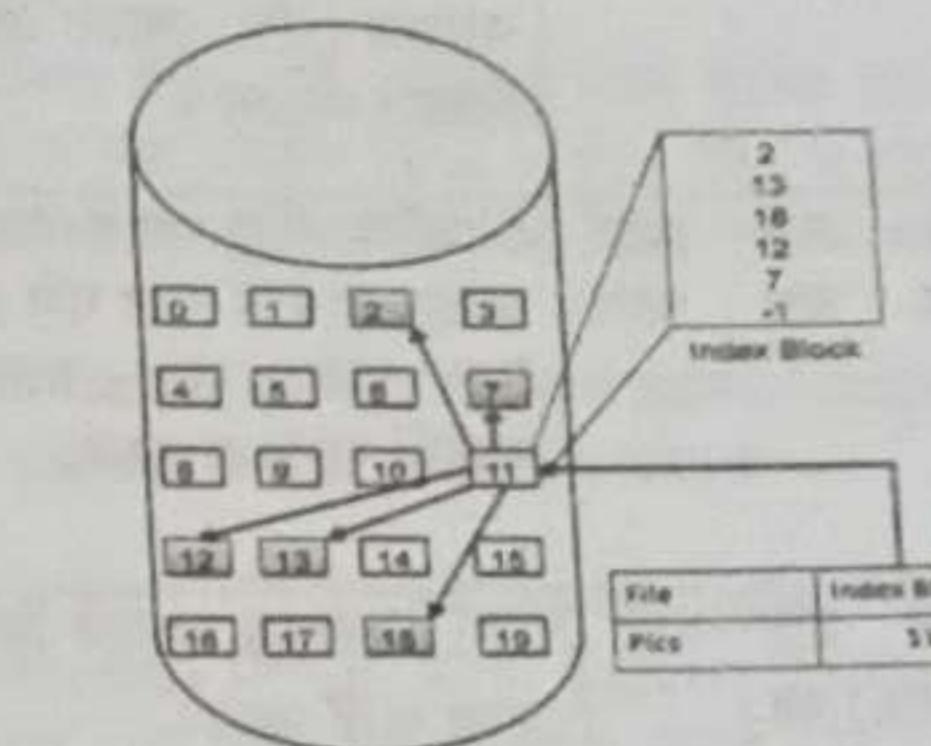


Figure: Indexed Allocation

চিত্রে দেখা যায় যে, Directory Index Block এর Address ধারন করে। Disk Space এর 'i'তম Block টি পড়তে চাইলে Pointer টি 'i'তম Index Block এ Set করতে হয় এবং পদ্ধতিটি অনেকটা Paging Scheme এর মতো। যখন কোন File তৈরি করা হয় তখন Index Block এর সর্বক প্রথম Pointer এর ব্যবহার করা হয়েছে। 'i' তম Block তিনে দেখা জন্য 'i'তম Index Block Entry তে এর Address লিখতে হয়।

**সুবিধা সমূহঃ**

- কোন External-Fragmentation সমস্যা নেই।
- Size-Declaration সমস্যা সমাধান করা যায়।
- Direct Access Support করে।

**অসুবিধা সমূহঃ**

- Indexed Allocation এ মেমোরি অপচয় বেশি হয়।
- Indexed Allocation এর Pointer Overhead Linked Allocation এর চেয়ে বড় হয়।

**প্রশ্ন-১৭. Allocation Methods of Disk Space/Disk** এর প্রেস বটন এর প্রযোজনীয়তা লিখুন।

উত্তর: Allocated Method of Disk Space এর প্রযোজনীয়তা: ডিক স্পেস কে বন্টন করলে ডিক স্পেস এর Perfect Use করা যায় এবং Disk সমূহকে দ্রুত নির্বাচন করা যায়।

**প্রশ্ন-১৮. Single Contiguous Allocation এবং Partition Allocation** এর মধ্যে পার্থক্য লিখ।

উত্তর: Single Contiguous Allocation এবং Partition Allocation এর মধ্যে নিম্নরূপ:

Single Contiguous Allocation	Partition Allocation
মেমোরি সময় অংশ ত্বক একটি	একই সাথে একাধিক প্রোগ্রাম নির্বাচন করা যায় বা হয়।
উক্ত প্রোগ্রাম নির্বাচনের কাজ শেষ হলে অন্য প্রোগ্রাম নির্বাচন নির্বাচন হয়।	

কোনো প্রোগ্রাম সম্পর্ক না হওয়া পর্যবেক্ষণ কৌশল সহজেই অনেক সহজ হয়। যদি প্রোগ্রাম কোনো প্রোগ্রামের নিচে নিচে থাকে।	CPU একই সময়ে একাধিক প্রোগ্রামক টাইম শারিংয়ে ভাগ করে নির্বাচন করতে পারে। যার ফলে সবসময় CPU কোনো প্রোগ্রামের নিচে থাকে।
প্রোগ্রামের দৈর্ঘ্য ছোট হলে মেমোরির বেশির ভাগ অশ্ব অববহত থাকে।	মেমোরির কিছু অশ্ব অববহত অববহত থাকতে পারে যদি প্রোগ্রাম লোড করার পরে এর প্রয়োজনীয় মেমোরির বরাদ্দ না থাকে।
CPU কে বেশির ভাগ সবই থাকতে আইডল (Idle) হয়।	CPU সবসময় প্রোগ্রাম নির্বাচনের কাজে ব্যর্ত থাকে।

প্রশ্ন-19. Fragmentation কি? ইহা কত প্রকার ও কি কিং বিভিন্ন প্রকল্প-২০১৬ সহজেই মেইনেল ইউনিভার্সিটি (মন.-কাউন্সেল)।  
উভয় Fragmentation পদের অর্থ খণ্ডন/খণ্ডকরণ। সাধারণত Memory Allocation এর সময় দুটি প্রোগ্রাম বা প্রসেসের মধ্যে কিছু ফর্ক অশ্ব সৃষ্টি হয়। Memory তে এই অববহতযোগ্য ফর্ক বা Fragments সৃষ্টি হওয়াকে Fragmentation বলে।  
ইহা দুই প্রকার। যথা:

- Internal Fragmentation
- External Fragmentation

#### প্রশ্ন-২০. Internal এবং External Fragmentation এর বর্ণনা দিখ।

উভয় Internal Fragmentation: যদি কখনো এমন হয় যে মেমোরিতে 50K জায়গা ফর্ক আছে এবং তাতে 49.9K এর একটি জব (Job) লোড করা হল। তাহলে .1K জায়গা অববহত থাকে, যা আর ব্যবহার করা যাব না। ইহাই Internal Fragmentation।

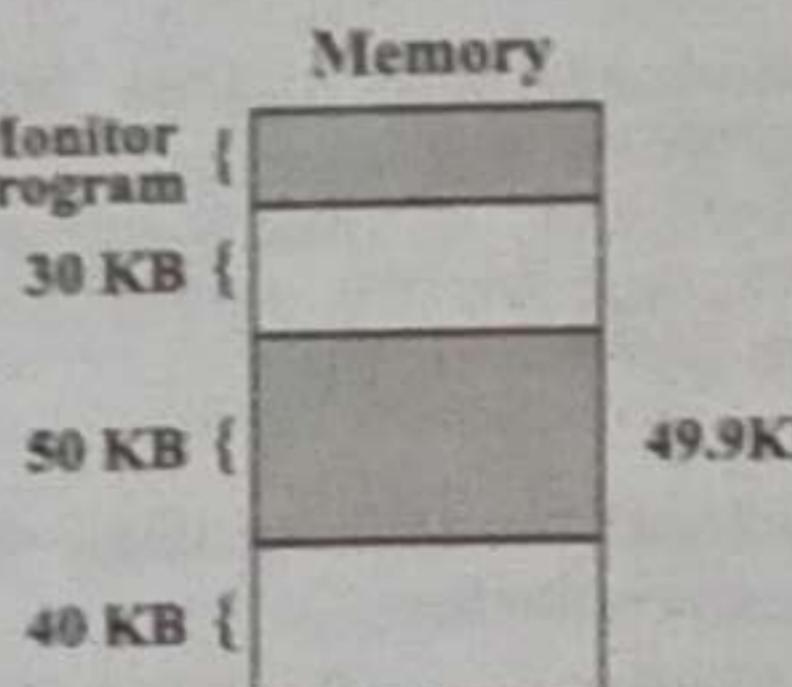


Figure: Internal Fragmentation

External Fragmentation: যদি কখনো এমন হয় যে যদি একটি অনেক এর Request এর জন্য যে পরিমাণ মেমোরি সরবরাহ তাৰ থেকে বেশি মেমোরি বিলামান কিং সেভলো ভিন্ন ভিন্ন জায়গায়। ফলে জব লোড করা যাব না তখন তাকে External Fragmentation বলে।

Swapping প্রক্রিয়ার বর্ণনা: অনেক ক্ষেত্রে অগ্রণ্যতা (Priority) ভিত্তিতে Swapping (সোরপিং) প্রক্রিয়া ঘটে। যদি কখনো মেমোরি মানেজার, Lower Priority প্রসেস নিয়ে কাজ করে। এমতাব্দী কোন Higher Priority প্রসেস যদি উপস্থিত হয়, তখন মেমোরি

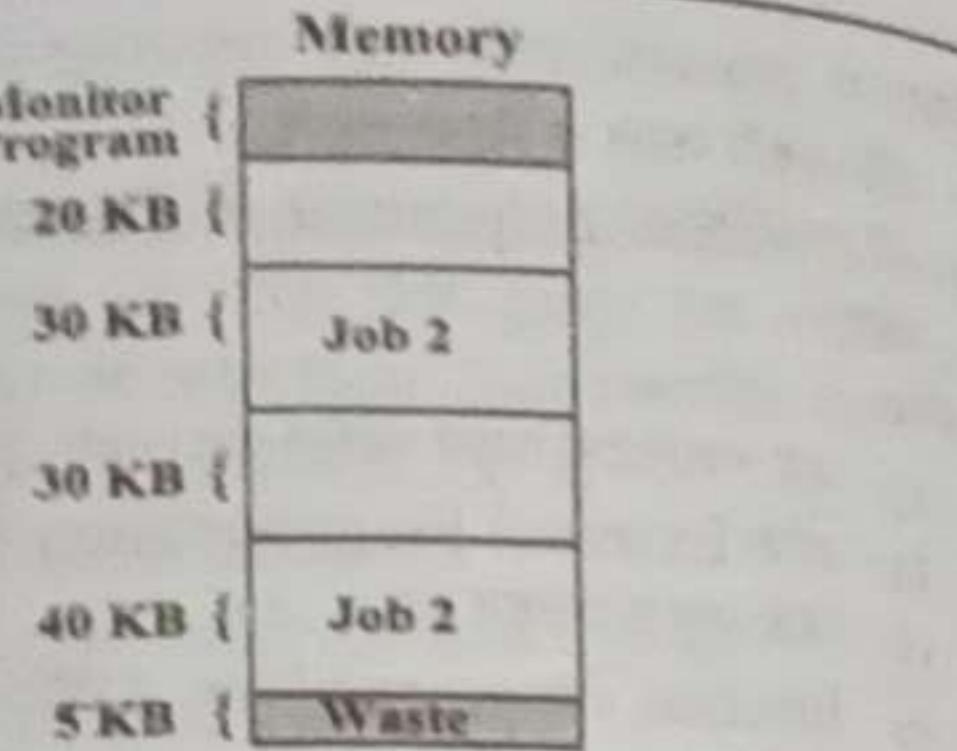


Figure: External Fragmentation

#### প্রশ্ন-২১. External Fragmentation দূর করার একটি উপায় দিখ।

উভয় External Fragmentation দূর করার জন্য Relocatable Partitioned Allocation পদ্ধতি ব্যবহার করা হয়।

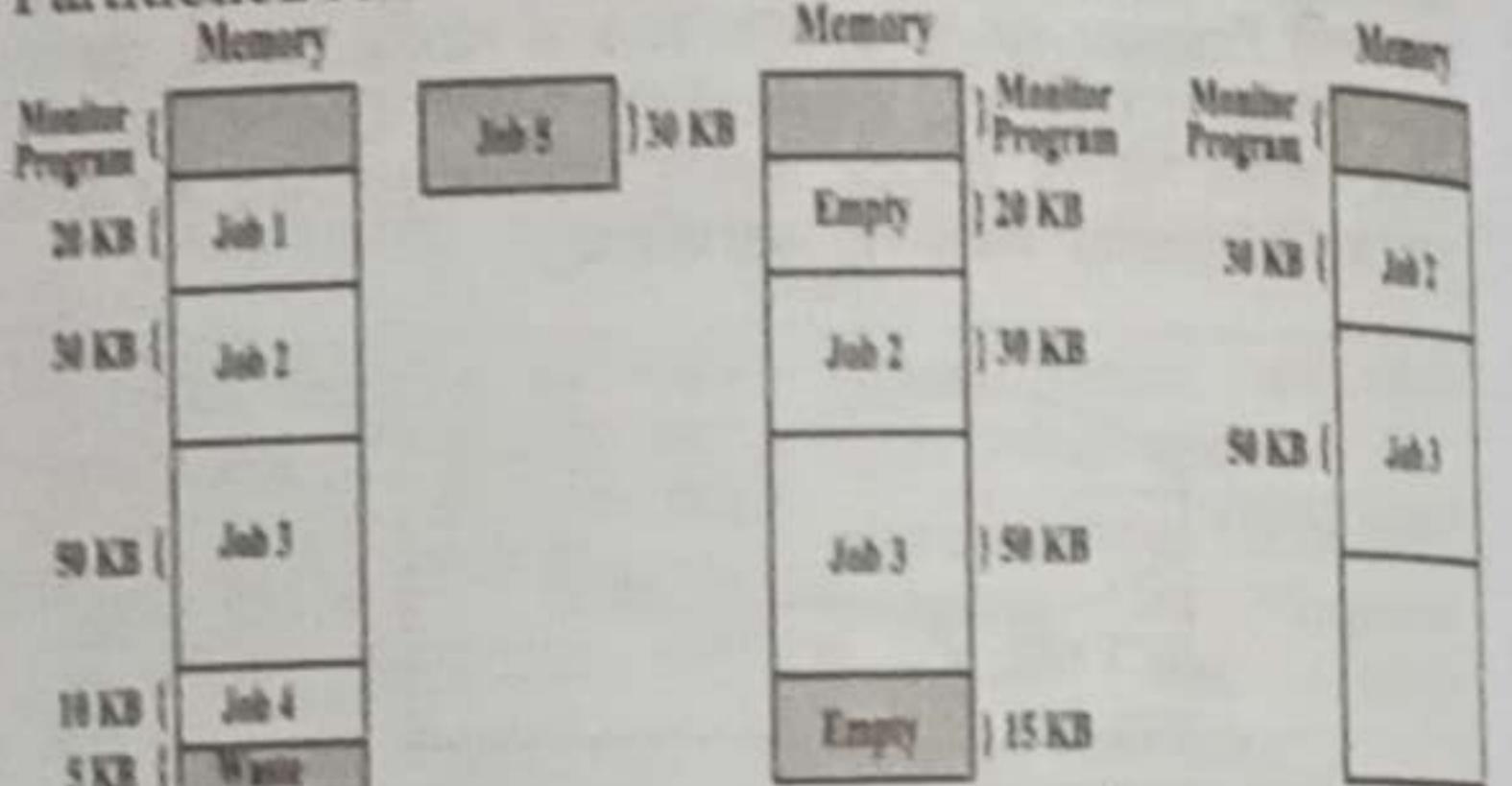


Figure: Relocatable Partitioned Allocation

- Initial
- Job1 & Job4
- Recompact Finished & load Job5

Relocatable Partitioned Allocation পদ্ধতিতে Fragmentation এর কারণে কোন জব লোড হতে বাধা প্রাপ্ত হল বিশেষ হার্ডওয়ার প্রয়োগ করে মেমোরি তে অবস্থিত বাকী সেল লোডেক থালি করে Requested জবকে মেমোরিতে লোড করে, যা উপরে চিহ্ন দেখানো হল।

#### প্রশ্ন-২২. Swapping কি? Swapping প্রক্রিয়া বর্ণনা করুন। [36<sup>th</sup> BCS, ICT Ministry - 12], [NTRCA(Lecturer)-2019]

উভয় Time Sharing System এ কিছু প্রোগ্রাম মেইন মেমোরিতে থাকে আবার কিছু প্রোগ্রাম ভিকে (Secondary Memory, Disk Storage, Backing Storage ইত্যাদি) রাখা হয়। Higher Priority জব মেইন মেমোরিতে আবার Lower Priority জবে ভিকে। তিক থেকে মেমোরিতে এবং সেই মেমোরি থেকে তিক এ হালন করার প্রক্রিয়াকে Swapping বলে।

Swapping প্রক্রিয়ার বর্ণনা: অনেক ক্ষেত্রে অগ্রণ্যতা (Priority) ভিত্তিতে Swapping (সোরপিং) প্রক্রিয়া ঘটে। যদি কখনো মেমোরি মানেজার, Lower Priority প্রসেস নিয়ে কাজ করে। এমতাব্দী কোন Higher Priority প্রসেস যদি উপস্থিত হয়, তখন মেমোরি

মানেজার, Lower Priority প্রসেসকে Swap করে এবং Higher Priority প্রসেসকে নির্বাচন করে। এই নির্বাচন শেষ হলেই সে আবার Lower Priority প্রসেসকে নির্বাচন করতে পারে। এধরণের Swapping প্রসেসকে আবার Roll Out এবং Roll in বলা হয়।

Swapping এর জন্য Backing Store বা Fast Disk Storage ব্যবহার করা হয়। আবার প্রসেস গুলো Backing Store এ একটি Ready Queue তে অবস্থান করে। যখন মেইন মেমোরিতে প্রসেস নির্বাচন শেষ হবে, তখন মেইন মেমোরি ডিস্প্যাচারকে (Dispatcher) পরবর্তী প্রসেস এর জন্য Call করে। ডিস্প্যাচার, মেইন মেমোরি থেকে নির্বাচিত প্রসেসকে Swap Out করে এবং পরবর্তী প্রসেস, মেইন মেমোরিতে Swap In করে। এভাবেই Swap In এবং Swap Out প্রক্রিয়া শাখায়ে Swapping প্রসেস চলতে থাকে।

(i) First Fit: এই পদ্ধতিতে কোন প্রসেস যেখানেই হিঁ পার্টিশন (Free Partition) পাবে (যা কিনা পর্যাপ্ত সাইজের), সেখানেই প্রসেসটি লোড হবে।

(ii) Best Fit: এই পদ্ধতিতে কোন প্রসেস সব্য Release হওয়া Hole বা অন্য যেকোন Hole এ লোড হয়। যা সবচেয়ে Minimum Hole কিংবা প্রসেসটি ধারণের মতো পর্যাপ্ত পরিমাণ জায়গা তাতে বিলামান থাকে।

(iii) Worst Fit: এই পদ্ধতিতে প্রসেস Largest Hole এ লোড হয়। তাই এতে মেমোরি Wast (অপচ্য) হয়।

নিচে চিত্রের মাধ্যমে Single Input Queue সহ Fixed Memory Partitioned দেখানো হলো: Figure:(X(a)) তে দেখা যাব যে, 2560K এর একটি মেমোরি আছে, যাতে Allocate হওয়ার জন্য ৫ টি Input Process (Job) Queue আকারে Waiting State এ আছে। আবার Figure (Y) তে দেখা যে, প্রথম মেমোরিকে কত গুলো নির্দিষ্ট সাইজে ভাগ করা হয়েছে। যা ৩ টি Hole এ ৩ টি প্রসেস (P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>) বিলামান (Figure:Y(a))। আবার Figure:Y(a) তে দেখা যাব যে, পর্যাপ্তমে P<sub>2</sub> Process Terminate হয়েছে। পরবর্তীতে Figure:Y(c) উক Hole টি দুটি ভাগ হয়েছে এবং একটি Hole এ P<sub>4</sub> নামক প্রসেস লোড হয়েছে। Figure:Y(d) হতে দেখা যাব যে, P<sub>1</sub> Terminate হয়েছে, যাতে পরবর্তীতে P<sub>5</sub> Figure:Y(e) লোড হয়েছে।

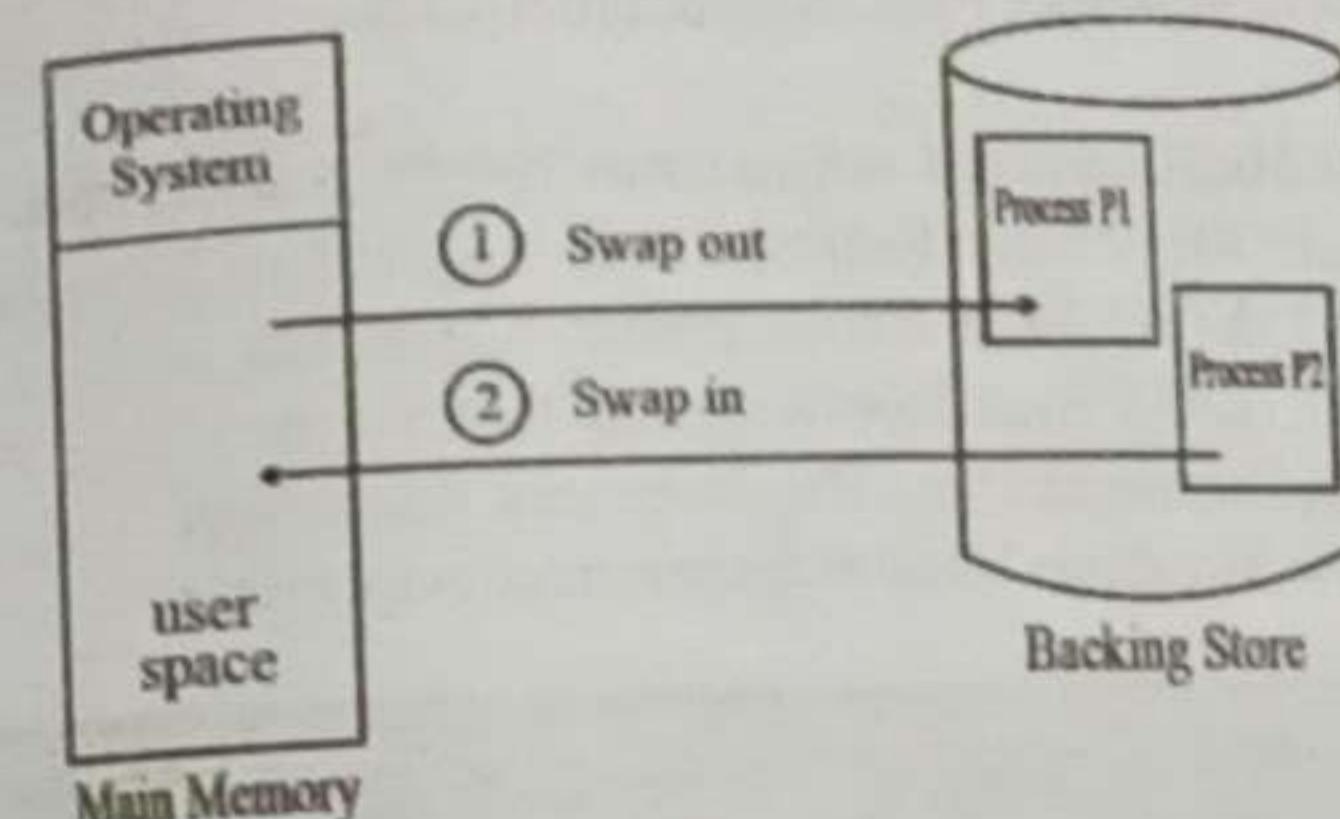


Figure: Swapping Process

#### প্রশ্ন-২৩. Partition Allocation এর সূবিধা ও অসূবিধা দিখ।

সূবিধা:

- Multiprogramming এর কারণে Partitioned allocation Schema ব্যবহারে idle time ছাপ পায়।
- Wasted Memory এর পরিমাণ ছাপ পায়।
- Throughput বৃক্ষি পায়।

অসূবিধা:

- বিশেষ Protection Mechanism এর প্রয়োজন হয়।
- অতিরিক্ত Core storage দরকার হয়।
- মেমোরি অববহত ব্যতিত অংশের উৎপত্তি ঘটে।

#### Figure(X): Single Input Queue সহ Fixed Memory Partition. a and b)

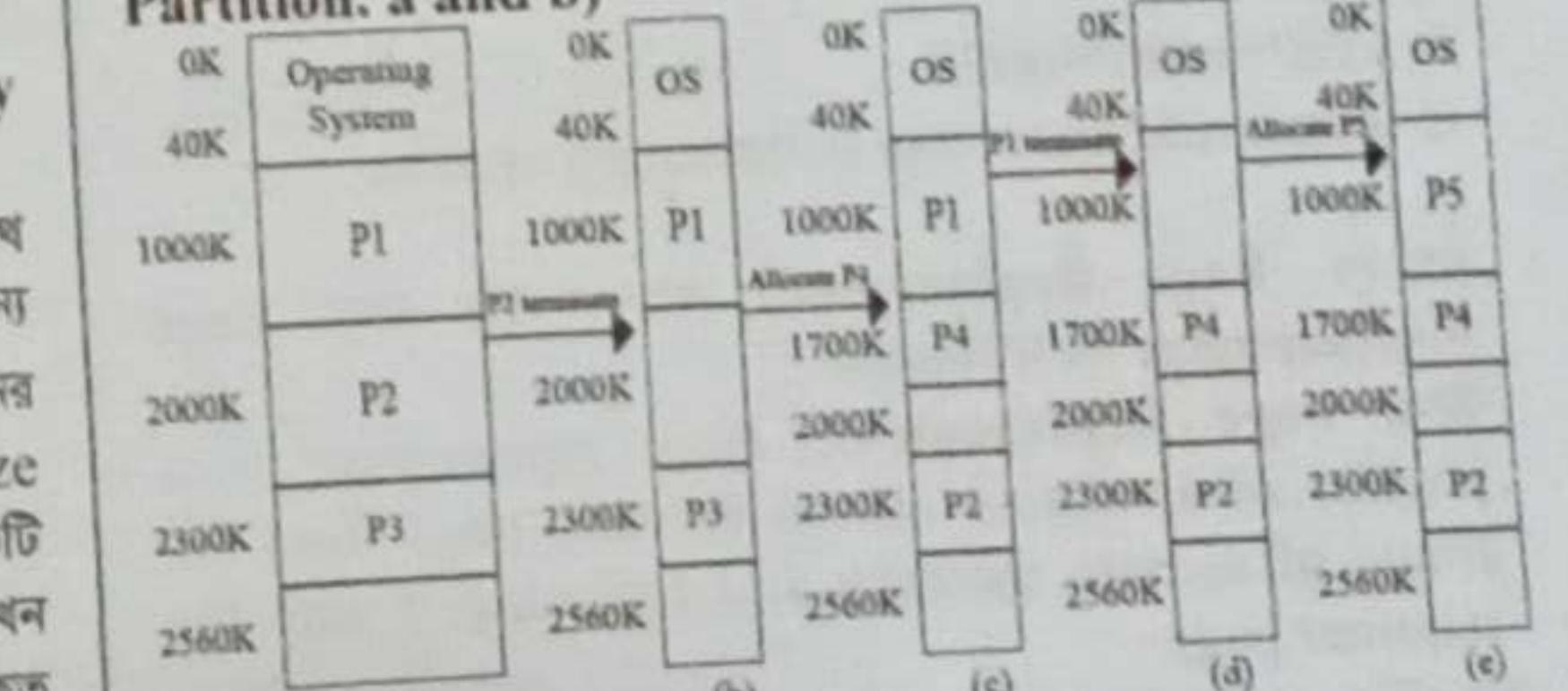


Figure (Y): Single Input Queue এর ক্ষেত্রে Fixed Memory Partitioned



তারপর আবার ০ আসবে, তখন আর পেজ ফল্ট হবে না, কারণ ০ অল্পেই পেজে আছে।  
তারপর আবার ৩ আসবে, তখন পেজ ফল্ট হবে, কারণ ৩ পেজে নাই, এখন ৭,০,১,২ এর মাঝে যে সবচেয়ে বেশি সময় পরে ব্যবহার হবে, বা হবেনা সে রিস্টেস হবে। তাই ৭ এর ঘরে রিস্টেস হবে। উপরের চিঠে দেখা যাচ্ছে ভবিষ্যতে (ভালিকে) আর ৭ ব্যবহার হচ্ছে।

তারপর আবার ০ আসবে, তখন আর পেজ ফল্ট হবে না, কারণ ০ অল্পেই পেজে আছে।

তারপর আবার ৪ আসবে, তখন পেজ ফল্ট হবে, কারণ ৪ পেজে নাই, এখন ২,১,০,৩ এর মাঝে যে সবচেয়ে বেশি সময় পরে ব্যবহার হবে, বা হবেনা, সে রিস্টেস হবে। তাই ১ এর ঘরে রিস্টেস হবে। উপরের চিঠে দেখা যাচ্ছে ভবিষ্যতে (ভালিকে) আর ১ ব্যবহার হচ্ছে। (২,০,৩ একটু পরেই ব্যবহার হবে।)

তারপর আবার ২,৩,০,২,০ ধারাবাহিক ভাবে আসবে, তখন আর পেজ ফল্ট হবে না, কারণ ২,০,৩,০,২,০ অল্পেই পেজে আছে।

Optimal page replacement is perfect, but not possible in practice as the operating system cannot know future requests.

#### সুবিধা সমূহ:

- a) Easy to Implement.
- b) Simple data structures are used.
- c) Highly efficient.

#### অসুবিধা সমূহ:

- a) Requires future knowledge of the program.
- b) Time-consuming.

**প্রশ্ন-৩:** নিচের Page Reference String তলো বিবেচনা করলে: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. নিচের Replacement Algorithms তলোর জন্য কতগুলো পেইজ ফল্ট স্টোরে, যদি- One, Two, Three, Four, Five, Six, Or Seven Frames হয়। মনে রাখতে হবে যে, সব frames প্রারম্ভিকভাবে বাতা (empty), সূতৰাং প্রতিটি পেইজ কন্টেন্টের সব Cost আপনার Unique (অনন্য) পেইজের হবে।

LRU replacement

FIFO replacement

Optimal replacement

উত্তর:

Number of frames	LRU Replace ment	FIFO Replace ment	Optimal Replace ment
1.	20	20	20
2.	18	18	15
3.	15	16	11
4.	10	14	8
5.	8	10	7
6.	7	10	7
7.	7	7	7

**প্রশ্ন-৪. Belady's Anomaly কি? ইয়ে ঘটার কারণ কী, আলোচনা করুন।**

**উত্তর:** Belady's Anomaly: সাধারণত, ভার্চুয়াল মেমোরির ক্ষেত্রে একটি প্রসেসরের জ্যেষ্ঠ সংখ্যা বৃদ্ধি, পেইজ ফল্ট এর সংখ্যা কম ঘটার চাহিতে ইহার Execution ক্রিয় বৃদ্ধি পায়। মাঝে মাঝে এর উক্সেটাইও হয়ে থাকে, যেমন: অধিক সংখ্যক পেইজ ফল্ট ঘটে যখন একটি অসেস অধিক পরিমাণে জ্যেষ্ঠ Allocated হয়। সবচেয়ে অপ্রত্যাশিত এই ঘটনাকলই Belady's Anomaly নামে পরিচিত।

(অন্যান্যে বলতে পেছে, Belady's Anomaly is the name given to the phenomenon where increasing the number of page frames results in an increase in the number of page faults for a given memory access pattern.)

**Belady's Anomaly ঘটার কারণ:** The other two commonly used page replacement algorithms are Optimal and LRU, but Belady's Anomaly can never occur in these algorithms for any reference string as they belong to a class of stack based page replacement algorithms.

A stack based algorithm is one for which it can be shown that the set of pages in memory for N frames is always a subset of the set of pages that would be in memory with N + 1 frames. For LRU replacement, the set of pages in memory would be the n most recently referenced pages. If the number of frames increases then these n pages will still be the most recently referenced and so, will still be in the memory. While in FIFO, if a page named b came into physical memory before a page – a then priority of replacement of b is greater than that of a, but this is not independent of the number of page frames and hence, FIFO does not follow a stack page replacement policy and therefore suffers Belady's Anomaly.

#### প্রারম্ভিক সিস্টেম:

**প্রশ্ন-৫. Process এর Aging Technique কি? কেন এটি OS এ ব্যবহৃত হয়।**

উত্তর: Aging হল একটি Technique যা সাধারণত Process এ Priority Increase করে যাহা অনেকক্ষণ ধরে System এর জন্য Wait করে থাকে। Starvation সমস্যা সমাধানের জন্য Aging Technique ব্যবহার করা হয়।

**প্রশ্ন-৬. Starvation কি?**

উত্তর: সেই অবস্থাকে Starvation বলে দেখানে Low Priority Processes তলোকে Block অবস্থায় পরে থাকে High Priority এর Process তলোকে প্রসেস করার কারণে।

#### ॥Secondary Storage Management॥

(Disk Scheduling Algorithms, I/O Systems)

**প্রশ্ন-৭. I/O System কাকে বলে?**

**উত্তর:** I/O System: যে পদ্ধতিতে কম্পিউটার এর ইনপুট ও অটকান্ত ডিভাইস এর সাহায্যে কম্পিউটার দ্বারা বিভিন্ন সমস্যা সমাধানের পথিকে বৃদ্ধি করা যায় তাকে I/O System বলে।

**প্রশ্ন-৮. I/O System -এ Hardware বলতে কি বুঝ?**

উত্তর: Computer এ ব্যবহৃত বিভিন্ন Device এর মধ্যে যে সকল Device Input Output Operation বা তর্হ সংরক্ষনের মাধ্যম হিসেবে ব্যবহৃত হয় সে সকল Device কে Hardware বলে।

**প্রশ্ন-৯. I/O System Software এর ক্ষেত্রে OS বিবেচ্য বিষয় সমূহ:**

- a) Data transfer mode
- b) Access method
- c) Sharing
- d) Device speed
- e) I/O Direction

এছাড়াও Operating System কে আরো কতগুলো বিষয়ের উপর লক রাখতে হয়, সেগুলো হলোঃ

- a) I/O Hardware এর Bus ব্যবহা।
- b) I/O Hardware Controller এর উপর।
- c) I/O ও তাদের রেজিস্টার সমূহের উপর।
- d) I/O hardware এর Interruption পদ্ধতির ক্ষেত্রে।

**প্রশ্ন-১০. Input/Output Hardware এর মূলনীতি চিহ্নিত কী?**

উত্তর: মূলত এর I/O Hardware বা Device সমূহ ক্যাবল বিবর Wireless এর মধ্যে Signal পাঠিয়ে Computer এর সাথে যোগাযোগ প্রতিষ্ঠা করে। একটি Computer System এর Fast Device (Processor Memory) সমূহ PCI Bus এবং Slow Device (Keyboard Mouse) সমূহ Expansion Bus এর সাথে সংযুক্ত থেকে I/O Hardware সমূহ পরিপন্থের সাথে যোগাযোগ প্রতিষ্ঠা করে। চিঠির মধ্যে কোন ক্ষেত্রেই নিয়ন্ত্রণের জন্য যে Controller রয়েছে তা এক বা একাধিক Register সমূহ Data ও Control Signal ধারণ করে এবং I/P ও O/P Port এর মধ্যে যোগাযোগ CPU Instruction এর পথে I/O এর Register Execute করে এবং Device Control Register এ R/W (Read/Write) Operation সম্পন্ন করে।

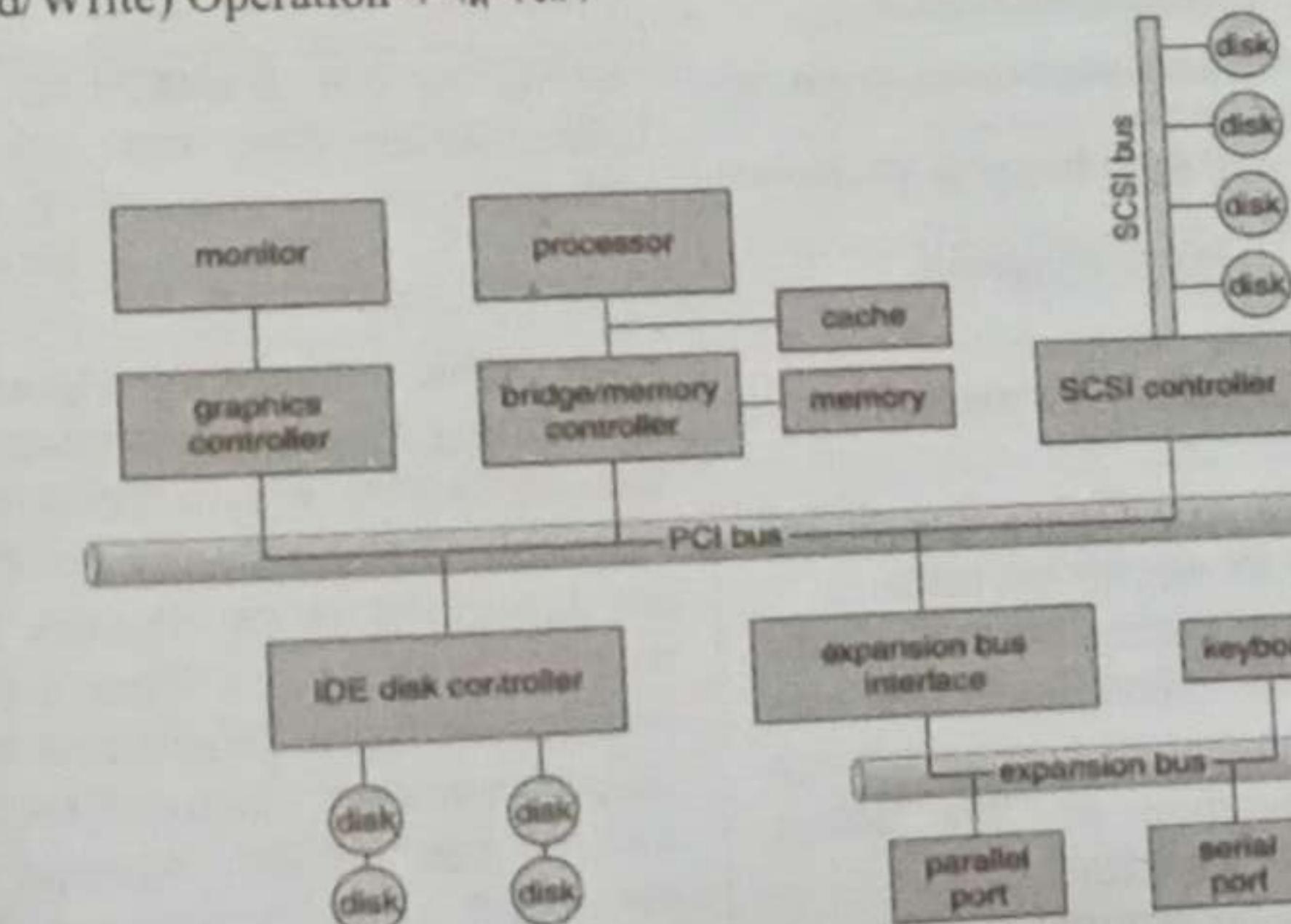


Figure: A typical PC bus Structure

প্রশ্ন-৭. I/O system এর প্রতিটি উভয়ের কার্যকলী/Function লিখ? উত্তর: নিম্ন Input/Output System এর জন্য তলো চিহ্নের সাহায্যে দেখানো হলো:

User Level I/O Software বা User Process
Device Independent O/S Software
Device Drivers
Interrupt Handlers
Hardware

নিম্ন Input/Output System এর জন্য তলোর কাজ চিহ্নের সাহায্যে দেখানো হলোঁ Input/Output System এর জন্য সমূহের প্রতিটি উভয়ের কার্যকলি উপরের চিহ্নে দেখানো হয়েছে। যার সর্বনিম্নে Hardware এরপরে ধাপে ধাপে Interrupt Handlers, Device Drivers, Device Independent Software এবং সর্ব উপরে User Process অবস্থান করছে। যা কিনা চিহ্নে তীব্র চিহ্ন (Flow of Control) দ্বারা সূচিত করা হয়েছে। যখন ব্যবহারকারী একটি ফাইল থেকে একটি Block পড়ার চেষ্টা করে, তখন অপারেটিং সিস্টেম সেটা করতে সাহায্য করে। Device Independent Software Block Cache রেজ করে এবং যদি Block টি দেখানো না থাকে, তবে এটি ডিভাইস ড্রাইভারের (Device Drivers) মাধ্যমে হার্ডওয়্যারের (Hardware) কাছে রিকুরেন্ট পাঠায়। যতক্ষণ পর্যন্ত ডিক অপারেশন (Disk Operation) সম্পন্ন না হয় ততক্ষণ প্রসেসটি Blocked অবস্থায় থাকে। যখন ডিস অপারেশন (Disk Operation) সম্পন্ন হয় তখন হার্ডওয়্যার একটি Interrupt Handlers থেকে বের করে যে, এই মুহূর্তে কোন ডিভাইস প্রয়োজন। তখন ইহা কন্ট্রোলার ডিভাইস থেকে অবস্থান জেনে নিয়ে সম্পন্ন করার জন্য প্রসেসটিকে জাগিয়ে তুলে এবং ব্যবহারকারীর প্রসেসটি এভাবেই চলতে থাকে।

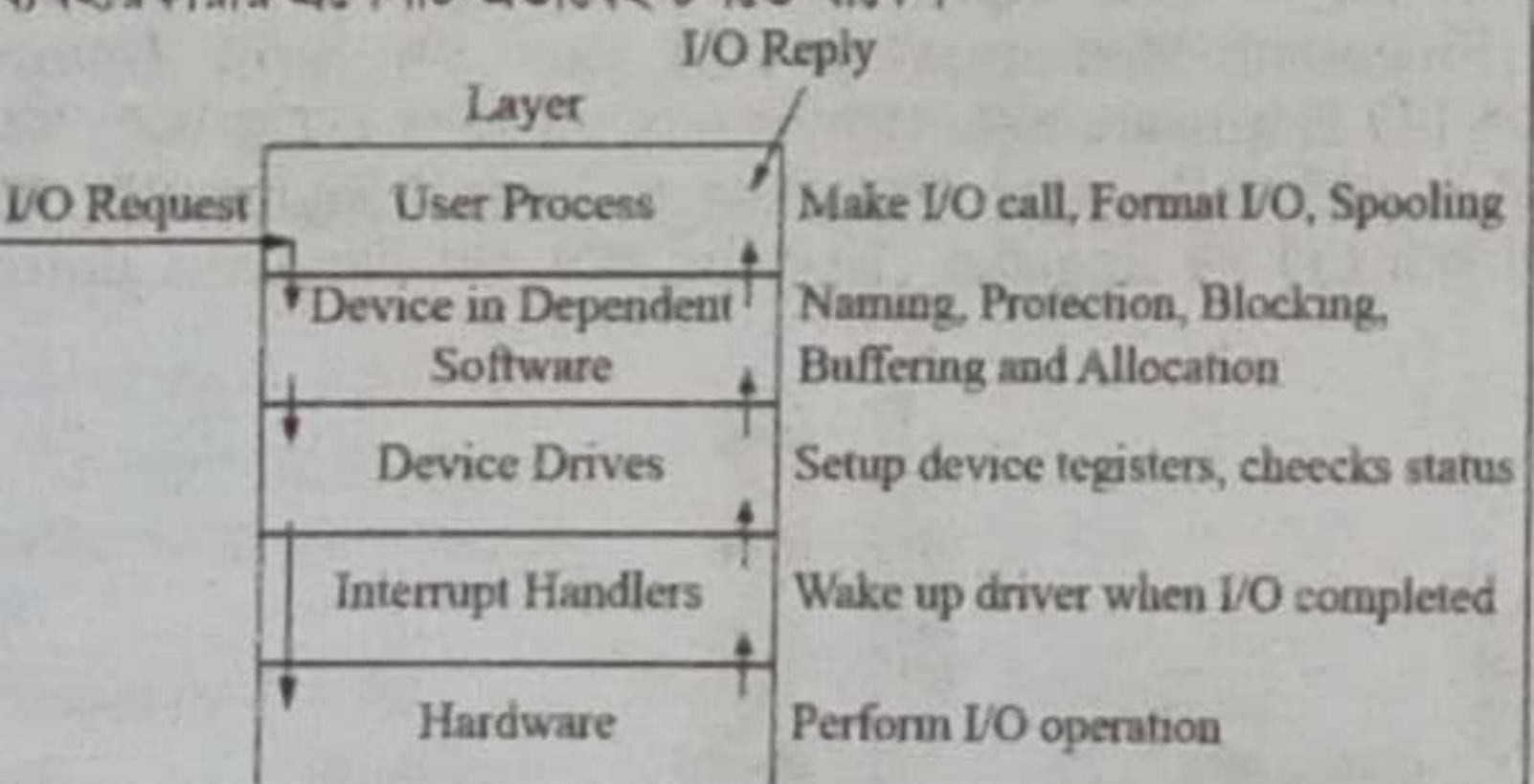


Figure: Input/Output System এর জন্য তলোর প্রধান কাজ

প্রশ্ন-৮. Storage Device কাকে বলে? Operation এর উপর ভিত্তি করে Storage Device কে কয় ভাগ করা হয়েছে।

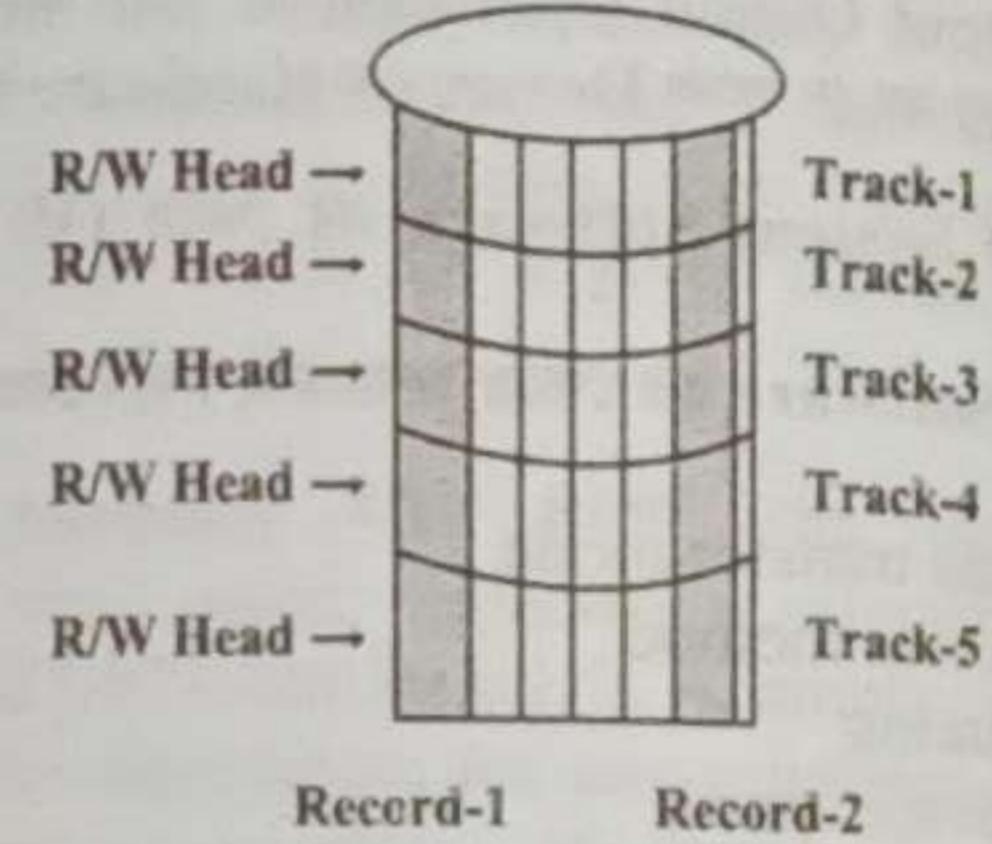
উত্তর: Storage Device: Storage Device হল এমন এক প্রকার মেক্যানিজম (Mechanism). যার সাহায্যে Input হতে পৃথীত Data কে Process করে সংরক্ষণ ও Output প্রদান করা যায় তাকে Storage Device বলে। Operation এর উপর ভিত্তিকরে Storage Device কে দুই ভাগে ভাগ করা হয়েছে:

a) Serial Access Storage Device (SASD)

### b) Direct Access Storage Device (DASD)

a) SASD: Serial Access Storage Device সমূহের সাহায্যে Data সমূহকে Serial Access করা হয়। যেমনঃ ম্যাগনেটিক টেপ।

b) DASD: Direct Access Storage Device এর কৈবল্য হচ্ছে এখানে Data Access করার সময় কোন ধারাবাহিকতা অবস্থায় করতে হয় না। অর্থাৎ যে কোন Location থেকে Data Access/Store করা যায়। যেমনঃ Magnetic Drum.



প্রশ্ন-৯. SASD & DASD এর মধ্যে পার্থক্য তলো লিখুন?

উত্তর: SASD & DASD এর মধ্যে পার্থক্য তলো লিখুন:

SASD	DASD
SASD হলো Serial Access Storage Device এর সংক্ষিপ্ত রূপ।	DASD হলো Direct Access Storage Device এর সংক্ষিপ্ত রূপ।
SASD এ তথ্য জমা করার জন্য অনুকূলিক অনুযায়ী প্রবেশ করতে হয়।	DASD এ এই অনুকূলিক নরকার হয়না।
SASD এ প্রথম হতে দূরের কোন রেকর্ডে যাওয়ার জন্য মধ্যবর্তী কিছু রেকর্ড এডিয়ে (Skip) হতে হয়।	DASD এ রেকর্ড এডিয়ে (Skip) যাওয়ার প্রয়োজন নেই।
SASD এ ডাটা রিড/গ্রাইট (Read/Write) করতে সময় বেশি লাগে।	DASD এ SASD এর তুলনায় ডাটা রিড/গ্রাইট (Read/Write) করতে সময় কম লাগে।

প্রশ্ন-১০. Disk Scheduling Algorithm কাকে বলে? Disk Scheduling Algorithm তলোর নাম লিখুন?

উত্তর: সবচেয়ে তালো অপারেটিং সিস্টেম হলো যা বিভিন্ন হার্ডওয়্যারকে Efficiently ব্যবহার করতে পারে এবং ইহা সঠিকভাবে করতে পারে। একটি Access করা যায় এবং বড় Disk Bandwidth প্রয়োজ্য। ম্যাগনেটিক ডিকের Access Time সাধারণত ডিক এর পজিশন ট্র্যান্সিউন্ড করে এবং Extra Time প্রয়োজন হয় ডিক ঘুরে নিদিষ্ট সেক্টর (Sector) যাওয়ার জন্য। আবার মোট সময়ে কি পরিমাণ Byte হারাব হলো তাই হচ্ছে ব্যাটউইথ। অনেকসময় ডিকের বিভিন্ন ইন্সেন্ট একসাথে ডেটা Access করার অনুরোধ (Request) আসতে পারে। কিন্তু কখন, কিভাবে, কোন অনুরোধ আগে Access করতে হবে তা

উপর ভিত্তি করে Access Time কমানো যায়। এর জন্য বিভিন্ন ধরণের ডিক শিডিউলিং এলগরিদম (Disk Scheduling Algorithm) ব্যবহার করা হয়।

যেমন:

- a) FCFS Scheduling
- b) SSTF Scheduling
- c) SCAN Scheduling
- d) C-SCAN Scheduling
- e) Look Scheduling
- f) C-Look Scheduling

প্রশ্ন-১১. Single-User System এ কিভাবে আমরা ফাইল নিরাপদ রাখতে পারি?

উত্তর: Single-User System এ ফাইল নিরাপদ রাখার উপায় সমূহ:

- a) Hide the disks.
- b) Use file names that can't be read.
- c) Backup disks.
- d) On floppies, place a write-disable-tab on.

প্রশ্ন-১২. RAID Structure কাকে বলে?

উত্তর: RAID Structure: Redundant Arrays of Independent Disks কে সংক্ষেপে RAID বলে। সাধারণত ডিক স্লাইট এখন আকারে ছোট ও দামে সত্ত্ব হওয়ায় একটি সিস্টেমে একসাথে অনেক ডিক সংযোগ করা যায়। যার ফলে ডেটা ট্রান্সফার এর ক্ষমতা বৃদ্ধি পায়। যদি কাজটা প্যারালালভাবে করা হয়। আবার একই ডেটা ভিত্তিক থাকলে কোন কারনে একটি নষ্ট হলে ডেটা হ্যারানোর ভয় থাকেন। বিভিন্ন ধরণের ডিক সংযোগের এই পক্ষতিকে একত্রে RAID Structure বলা হয়।

প্রশ্ন-১৪: দেওয়া আছে, ধারাবাহিকভাবে 100K, 500K, 200K, 300K, এবং 600K মেমোরি পার্টিশন। কিভাবে First-fit, Best-fit, এবং Worst-fit এলগরিদম তলো 212K, 417K, 112K, এবং 426K (যথাক্রমে) জায়গা তলো প্রসেস করবে? কোন এলগরিদমটি মেমোরি ব্যবহারের ক্ষেত্রে সবচেয়ে বেশি কার্যকর?

উত্তর:

- First-fit:  
a) 212K is put in 500K partition
- b) 417K is put in 600K partition
- c) 112K is put in 288K partition (New partition  
288K = 500K - 212K)
- d) 426K must wait

Best-fit:

- a) 212K is put in 300K partition
- b) 417K is put in 500K partition
- c) 112K is put in 200K partition
- d) 426K is put in 600K partition

Worst-fit:

- a) 212K is put in 600K partition
- b) 417K is put in 500K partition
- c) 112K is put in 388K partition
- d) 426K must wait

সুরাগ বলা যাচ্ছে যে, উপরের উদাহরণে মেমোরি প্রসেসিং এর ক্ষেত্রে Best-fit এলগরিদমটি মেমোরি ব্যবহারের ক্ষেত্রে সবচেয়ে বেশি কার্যকর।

প্রশ্ন-১৫: একটি সিস্টেম বিবেচনা করলে দেখানো একটি প্রয়োজ কোড এবং ডাটা এই দুটি আলাদা ভাগে বিভক্ত থাকে। CPU জানে যে ইহা হয় একটি নির্দেশনা (Instruction) এবং অথবা ডাটা (Data)। অন্যদিক থেকে একটি প্রয়োজ কোড করা হয়: যার একটি ইনস্ট্রিক্ষন বিভিন্ন ধরণের ব্যবহারকারীদের মধ্যে শেয়ার করতে পারে। এই পক্ষতির সুবিধা ও অসুবিধা আলোচনা করুন।

উত্তর: এই পক্ষতির মুখ্য সুবিধা হচ্ছে- ডেটা ও কোড শেয়ারিং এর জন্য ইহা কার্যকর পক্ষতি। উদাহরণ ব্যক্ত, একটি এভিটের ইনস্ট্রিক্ষন (Instruction Fetch) অথবা ডাটা (Data Fetch Or Store) চায়। অন্যদিক থেকে একটি ইনস্ট্রিক্ষনের জন্য অপরটি ডাটার জন্য অসুবিধা আলোচনা করুন। ইহার অন্য একটি সুবিধা হচ্ছে, কোডকে ভুল পরিবর্তন থেকে ইহা নিরাপদ রাখে।

এই পক্ষতির একটি মাত্র অসুবিধা হচ্ছে, কোড ও ডেটা অবশ্যই আলাদা থাকতে হয়, যা সাধারণত কম্পাইলার জেনারেটেড কোড এর সঙ্গে থাকে।

প্রশ্ন-১৬: পেইজের সাইজ সংস্থার পাওয়ার ২ হয় কেন? উত্তর: আমরা জানি, পেইজিং ব্যবহারিত হয় একটি এন্ডেসকে একটি পেইজ ও অফসেট নামের ভাগ করে। বরং পেইজ নামের এবং অফসেট ক্যালকুলেট করার জন্য এন্ডেসটিক পারফর্ম করার পরিবর্তে এন্ডেসকে পেইজ এবং অফসেট বিটস এ ভাঙা খুবই কার্যকর। কারণ প্রত্যেকটা বিট পজিশন প্রকাশ করা হয় Power of 2 দ্বারা, একটি এন্ডেসকে বিট রেজাস্টের মধ্যে ভাগ করা হয় পেইজ সাইজের মধ্যে এর Power of 2 হয়।

প্রশ্ন-১৭: নিচোড় পেইজ রিপ্রেসেন্ট এলগরিদম গোলো বিবেচনা করুন। পেইজ ভলোর পেইজ ফাঁট হার “Bad” to “Perfect” অনুসারে ফাইল পয়েন্ট কেলে তাদের রাঙ্ক (Rank) বের করুন। বেসকল এলগরিদম গোলো Belady's Anomaly Suffer করে এবং বেসকল Suffer করে না তাদেরকে অলাদা করুন।

- a. LRU replacement
- b. FIFO replacement
- c. Optimal replacement
- d. Second-chance replacement

উত্তর:

Rank	Algorithm	Suffer from Belady's Anomaly
1.	Optimal	No
2.	LRU	No
3.	Second-chance	Yes
4.	FIFO	Yes

প্রশ্ন-১৮: ১ মাইক্রোসেকেন্ড এর একটি সাইকেল টাইমসহ একটি সেক্ষ্ট্রাল প্রসেসর ব্যবহার করা একটি অপারেটিং সিস্টেম একটি Paged Virtual Memory সাপোর্ট করে। চলমান একটি পেইজের এর চাইতে অতিরিক্ত ১ মাইক্রোসেকেন্ড সময় ব্যবহার করে অন্য একটি পেইজ এন্ডেস করতে। পেইজে ১০০০ শব্দ আছে এবং পেইজিং ডিভাইস হচ্ছে একটি ড্রাম (Drum) ইহা প্রতি মিনিটে ৩০০০ রিভোলিশন (Revolutions) রোটেট (Rotate) ও প্রতি সেকেন্ডে ১ মিলিয়ন শব্দ ট্রান্সফার করতে পারে। নিম্নলিখিত সিস্টেম থেকে পরিবাপ্ত করে পাওয়া যায়।

- সকল ইন্ট্রাকশনের ১ শতাংশ এন্ড্রিভিটেড এন্ডেসেড করে অন্য একটি চলমান পেইজের চাইতে।
- অন্য পেইজ এন্ডেসেড করা ইন্ট্রাকশন, ইন্ট্রাখার্ডে একটি পেইজের ৮০ পারসেট এন্ডেসেড মেমোরিতে আছে।
- যখন একটি নতুন পেইজ দরকার হিলো, রিপ্রেসেড পেইজ সময়ের ৫০% মডিফাই করেছে।

প্রশ্ন-১৯: এই সিস্টেমের Effective Instruction Time ক্যালকুলেট করুন, মনে করুন, এই সিস্টেমের শুরুমাত্র একটি প্রসেস রানিং আছে এবং ড্রাম ট্রান্সফার চলাকালীন প্রসেসরস আইডল অবস্থায় আছে।

সমাধান: 
$$\text{Effective instruction time} = 0.99 \times (1 \mu\text{sec} + 0.008 \times (2 \mu\text{sec}) + 0.002 \times (10,000 \mu\text{sec} + 1,000 \mu\text{sec}) + 0.001 \times (10,000 \mu\text{sec} + 1,000 \mu\text{sec})) = (0.99 + 0.016 + 22.0 + 11.0) \mu\text{sec}$$

= 34.0  $\mu\text{sec}$

প্রশ্ন-২০: ডাইরেক্ট এন্ডেস ফাইল স্ট্রাট এন্ডেস করার ক্ষেত্রে এগুলো ইনডেক্স ক্যালকুলেট করার জন্য এন্ডেসটিক পারফর্ম করার পরিবর্তে এন্ডেসকে পেইজ এবং অফসেট বিটস এ ভাঙা খুবই কার্যকর। কারণ প্রত্যেকটা বিট পজিশন প্রকাশ করা হয় Power of 2 দ্বারা, একটি এন্ডেসকে বিট রেজাস্টের মধ্যে ভাগ করা হয় পেইজ সাইজের মধ্যে এর Power of 2 হয়।

প্রশ্ন-২১: ISAM কি? আলোচনা করুন।

উত্তর: Indexed sequential access method কে সহজেই ISAM বলে। শার্ট অর্ডারের ক্ষেত্রে ফাইল ভলো শর্টেড হয়ে থাকে। ISAM এর একটি মাস্টার ইনডেক্স ফাইল (Master Index file) আছে, যা কিনা অন্যকোন ইনডেক্স ফাইলের কোন কি (Key) টা হারি ঢাঁচেন ফাইল রেকর্ডসের সেকেন্ডারি ইনডেক্স পয়েন্টস (Secondary index) থেকে তা নির্দেশ করে উভয়ক্ষেত্রে রেকর্ড লোকেট (Locate) করা জন্য Binary Search ব্যবহার করা হয়।

প্রশ্ন-২২: এমন চারটি উপায় লিখুন যেখানে সিস্টেম কোন ব্যবহারকারী ফাইলকে অন্যান্য ব্যবহারকারীদের থেকে নিরাপদ রাখতে পারে। (List four ways systems might provide for users to protect their files against other users.)

উত্তর: কোন ব্যবহারকারীর ফাইলকে অন্যান্য ব্যবহারকারীদের থেকে নিরাপদ রাখার চারটি উপায় নিম্নরূপঃ

- ব্যবহারকারীদের ফাইল নামকরণে unprintable characters ব্যবহার করতে হবে, যাতে অন্যান্য ব্যবহারকারীরা সম্পূর্ণ নামটি নির্ধারণ করতে না পারে।
- আর্যাজের অনুমতি দেওয়ার আগে অবশ্যই প্রতিটি ফাইলে পাসওয়ার্ড প্রদান করা উচিত।
- অ্যারেস তালিকা নির্ধারণ করে, প্রতিটি ফাইল ব্যবহার করে অনুমতিপ্রাপ্ত প্রত্যেককে তালিকাভুক্ত করা।
- প্রতিটি ফাইলে সুরক্ষা কোড ব্যবহার করা জন্য, ব্যবহারকারীদের ধরন অনুসারে প্রেরণ করা।

প্রশ্ন-২৩: Disk Scheduling এ Single-User Environment এর মধ্যে একটি FCFS Scheduling ব্যাক্তি অন্যান্য পিচিটি কি কার্যকর? আলোচনা করুন। (Is disk scheduling, other than FCFS scheduling, useful in a single-user environment? Explain your answer.)

উত্তর: Single-User Environment এর মধ্যে, সাধারণত I/O Queue থাকে। সাধারণত একটি Block এর জন্য অথবা একটি ধারাবাহিক Block sequence এর জন্য Requests একটি সিলেক্স প্রসেস থেকে পৌছায়। এই ক্ষেত্রে, Disk Scheduling এর জন্য FCFS Scheduling হচ্ছে একটি সাধারণ পদ্ধতি। কিন্তু LOOK Scheduling প্রোগ্রাম হিসেবে প্রায়ই সহজ এবং অনেক জাতে পারফর্মেস প্রদান করে যখন মাল্টিটাইল প্রসেস একইসঙ্গে I/O হিসেবে পারফর্ম করে, যেখন যখন একটি উয়েব ব্রাউজার ব্যাক ফাউন্ডেশন থেকে ডেটা পুনরুৎসব করে তখন ফোর্মাটেড (Foreground) অপারেটিং সিস্টেম পেইজিং এবং অন্যান্য Application এর থাকে।

DRAM = Dynamic Random Access Memory.

DASD = Direct Access Storage Device.

RASD = Random Access Storage Device.

PWD = Print work directory.

Mkdir = Make directory.

CAT = Concatenate.

PCI = Peripheral Component Interconnect.

ISA = Industry Standard Architecture.

প্রশ্ন-২৪: ডিক্টিউলিং এলগরিদম নির্বাচনের ক্ষেত্রে আপনি কিভাবে র্যাম ডিক্ট এফেক্ট ব্যবহার করবেন? কোন কারণ ভলো আপনার বিবেচনা করা প্রয়োজন? হার্ড ডিক্টিউলিং এর ক্ষেত্রেও একটি ধরনের বিবেচনা এগুলো করুন।

প্রশ্ন-২৫: স্ট্যাবল স্টোরেজ (Stable Storage) ব্যবহার করার কোন উপায় আছে কি? আলোচনা করুন।

উত্তর: স্ট্যাকারের স্ট্যাবল স্টোরেজ (Stable Storage) থেকে ডাটা কখনে হ্যারায়ন। স্ট্যাবল স্টোরেজ এর জন্য মৌলিক টেকনিক হচ্ছে ডাটাকে একাধিক জায়গায় কপি করে রাখা, যাতে করে যদি ডাটার কোন কপি মুছে যায় বা ক্ষণ হয়ে যায়, অন্যান্য কপি ব্যবহার করার জন্য একেইলেক্স থাকে। কিন্তু কোন প্রক্লেভের জন্য, আমরা একটি বড় ধরনের দুর্ঘটনার কথা কল্পনা করতে পারি, যাতে সব কপি ডাটা ক্ষণ হয়ে যাব বা মুছে যায়।

প্রশ্ন-২৬: স্ট্যাবল স্টোরেজ (Stable Storage) ব্যবহার করার কোন উপায় আছে কি? আলোচনা করুন।

উত্তর: স্ট্যাকারের স্ট্যাবল স্টোরেজ (Stable Storage) থেকে ডাটা কখনে হ্যারায়ন। স্ট্যাবল স্টোরেজ এর জন্য মৌলিক টেকনিক হচ্ছে ডাটাকে একাধিক জায়গায় কপি করে রাখা, যাতে করে যদি ডাটার কোন কপি মুছে যায় বা ক্ষণ হয়ে যায়, অন্যান্য কপি ব্যবহার করার জন্য একেইলেক্স থাকে। কিন্তু কোন প্রক্লেভের জন্য, আমরা একটি বড় ধরনের দুর্ঘটনার কথা কল্পনা করতে পারি, যাতে সব কপি ডাটা ক্ষণ হয়ে যাব বা মুছে যায়।

প্রশ্ন-২৭: স্লো স্টোরেজ (Slow Storage) এর ক্ষেত্রে স্লো ডিভাইস কি? আলোচনা করুন।

উত্তর: স্লো স্টোরেজ (Slow Storage) এর ক্ষেত্রে স্লো ডিভাইস কি? আলোচনা করুন।

(i). Processor. (ii). Memory. (iii). Cache Memory

Slow Device নাইট হলঃ

(i). Keyboard. (ii). Mouse

প্রশ্ন-২৮: I/O Hardware এর Basic Elements কি? আলোচনা করুন।

উত্তর: I/O Hardware এর Basic Elements কি? আলোচনা করুন।

(i). Port (ii). Bus (iii). Device Controller

প্রশ্ন-২৯: I/O Hardware সমূহের Speed Range কত?

উত্তর: I/O Hardware বা Device সমূহের Speed Byte Per Second থেকে Gigabyte Per Second পর্যন্ত হতে পারে।

প্রশ্ন-৩০: Synchronous Device কি বুঝ?

উত্তর: যে সকল I/O Hardware নির্দিষ্ট সময় অন্তর Data Transfer করে তাদেরকে Synchronous Device বলে।

প্রশ্ন-৩১: Asynchronous Device কি?

উত্তর: যে সকল Device অনিয়ন্ত্রিত বিবরিতে Data Transfer করে তাদেরকে Asynchronous Device বলে।

প্রশ্ন-৩২: Computer এ ব্যবহৃত Fast ও Slow Device সমূহ কি? আলোচনা করুন।

উত্তর: Fast Device সমূহ PCI Bus এর সাথে এবং Slow Device সমূহ Expansion Bus এ সাথে সংযুক্ত থাকে।

প্রশ্ন-৩৩: I/O Software কি?

উত্তর: I/O Software হচ্ছে I/O Device সমূহের ব্যবহাগনা ও নিয়ন্ত্রণের জন্য প্রযোজনীয় Program সমষ্টি।

### File Management

(File System Structure, Organization, FCB, Space Allocation, Tree Structure files System)

প্রশ্ন-৩৪: File কি?

উত্তর: প্রযোজন

প্রশ্ন-২. বিভিন্ন ধরনের ফাইল সিস্টেম তালো লিখ।  
উত্তর: বিভিন্ন ধরনের ফাইল সিস্টেম তালো হলো:

- a). Disk File System
- b). Flash File System
- c). Database File System
- d). Transaction File System
- e). Network File System
- f). Special Purpose File System

প্রশ্ন-৩. File এর Attribute তালো কি কি?

উত্তর: File এর Attribute তালো হলো:

- (i) Name: অভ্যন্তরীন ফাইলের একটি নাম থাকে যা ব্যবহারকারী কর্তৃক প্রস্তুত।
- (ii) Type: File বিভিন্ন ধরনের হতে পারে। যেমনঃ Text File, Object File, Source File ইত্যাদি।

- (iii) Location: File টি কিন্তু এর কোন অংশে অবস্থিত তার Information ধারণ করে।
- (iv) Size: এটি ফাইল এর সাইজ বুকতে ব্যবহৃত হয়। অর্থাৎ ফাইল টি কত বাইট এর কিংবা Total File Space কত ইত্যাদি Information ধারণ করে।

- (v) Protection: এই অংশের File Protection এর বিভিন্ন Information জানা যায়।
- (vi) Time Date And User Identification: এই অংশে ফাইলটি কখন তৈরি হয়েছে বিংবা মডিফাই করা হয়েছে এবং এর ইউজার সম্পর্কিত বিভিন্ন Information জানা যায়।

vii. Password: ফাইলকে একসেস করার জন্য প্রয়োজনীয় পাসওয়ার্ড।

- viii. Record Length: Record Length হচ্ছে প্রত্যেক রেকর্ডের প্রয়োজনীয় বাইটের পরিমাণ।
- ix. Various Flag: ফাইলের বিভিন্ন ফ্লাগ রয়েছে। যেমনঃ Read Only, Hidden, Write Only Random Access, Archive Lock ইত্যাদি।

প্রশ্ন-৪. বৈশিষ্ট্য সহ File এর প্রকারভেদ লিখুন।

উত্তর: বৈশিষ্ট্য সহ File এর প্রকারভেদ হলো:

File Type	Extension	কাজ / Function
Executable	.exe, .com, .ben	Read to machine language program
Objective	.Obj, .O	Complied Machine language
Source Code	.C, .P, .Par, .asm, .a	Source code in various languages
Batch	.bat, .Sh	commands to the command interpreter
Text	.txt, .doc, .docx, .xls, .xlsx	Textual data document.
Library	.lib, .a	libraries of routines for programmers
Print or View	.ps, .dvi, .git, .xps	printing or viewing
Archive	.zip, .rar, .tar	for archiving or storage
Multimedia	.mpeg, .mov, .mpe, .mp4, .avi	Binary file containing audio or A/V information

প্রশ্ন-৫. File এর মৌলিক / Basic Operation সমূহ লিখুন।  
উত্তর: Basic File operation সমূহ হলো:

- > Create : নতুন ফাইল তৈরি করা।
- > Open: পুরাতন ফাইল উন্মোক্ত করা।
- > Read : ফাইল থেকে ডাটা পড়া।
- > Write: ফাইল এ ডাটা লিখা।
- > Close: ফাইল বন্ধ করা।
- > Delete ফাইল মুছে ফেলা।
- > Rename: ফাইলের নাম পরিবর্তন।
- > Append: সংযোজন করা।

প্রশ্ন-৬. রিসোর্স ব্যবহারে নিমিত্তে ধারাবাহিক অপারেশন তালো কি কি?  
উত্তর: রিসোর্স ব্যবহারে নিমিত্তে ধারাবাহিক অপারেশন তালো হলো:

- a) Request
- b) Use
- c) Release

প্রশ্ন-৭. Organization (সংগঠন) of file System / Logical Phases / File System এর Organization অক্ষয় করুন।

উত্তর: Organization (সংগঠন) of File System:

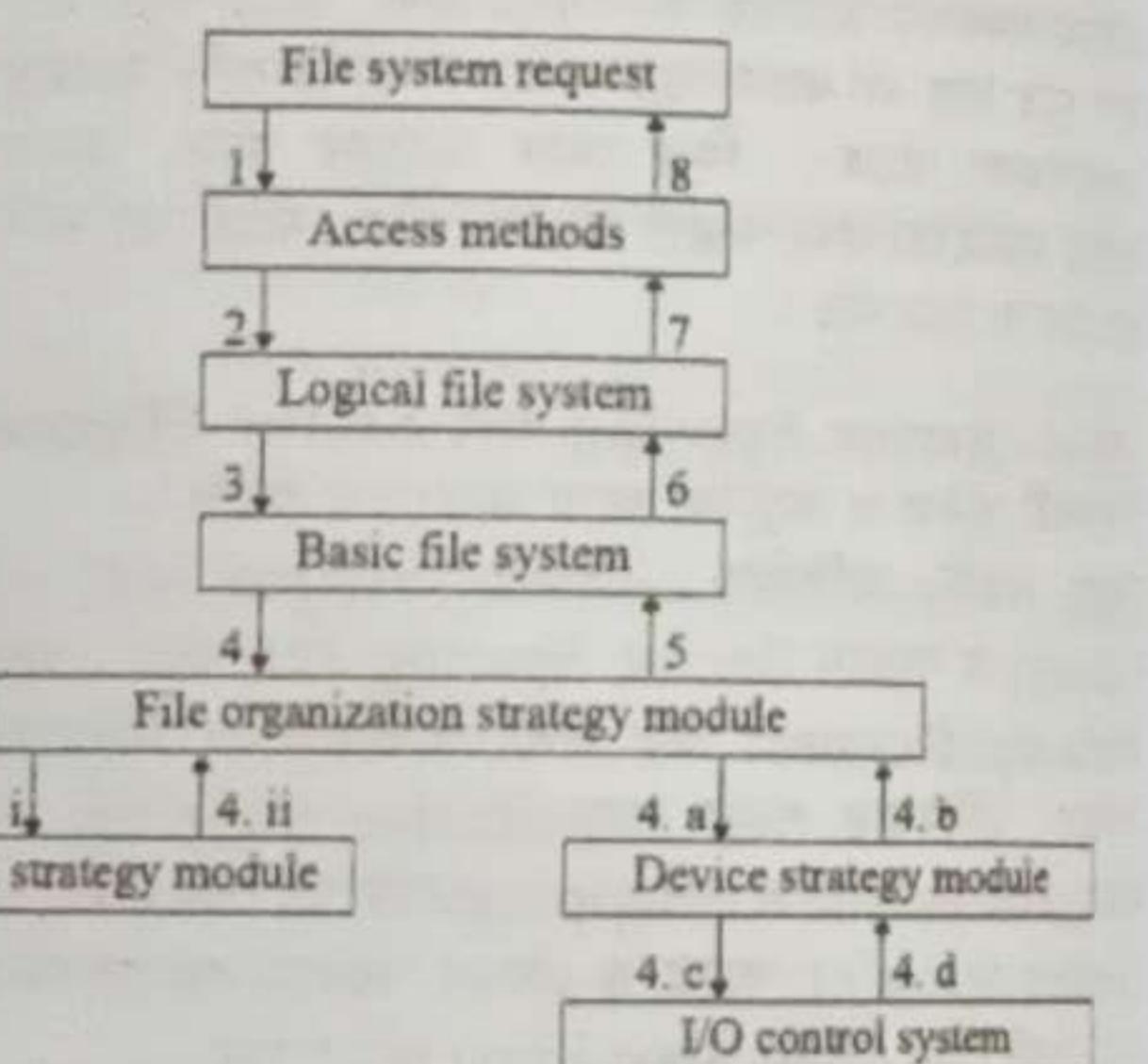


Figure: Organization (সংগঠন) of File System

System Organization: File এর Operation যে সকল Step এর সময়ে সম্পন্ন হয় তাদের ধারাবাহিক সময়ের কে Organization বলে। ফাইল সিস্টেম অর্গানাইজেশন একটি Multilevel Process (বহুধাপিক প্রক্রিয়া)। যার প্রত্যেকটি ধাপ অনেকের ক্ষেত্রে প্রযোজ্য। সাধারণ ফাইল সিস্টেমকে Process Sequence অনুসরে ৭ টি লজিকাল ধাপে ভাগ করা হয়েছে। নিচে প্রত্যেকটি ধাপের আলোচনা করা হলো।

(i) Access Method: এ পর্যায়ে ফাইলের লজিকাল রেকর্ড নামকে লজিকাল বাইট এক্সেসে রূপান্তর করে। এরপর Control কে লজিকাল ফাইল সিস্টেমে পাঠানো হয়।

(ii) Logical File System: এ পর্যায়ে Symbolic ফাইল নামকে এর অনুরূপ (Corresponding) VTOC (Volume Table of Content) Entry নামকে রূপান্তর করে। যদি Active Name Table (ANT) এ নাম পাওয়া না যায় তাহলে Master Directory & Sub Directory এর মাধ্যমে খুঁজে বের করা হয়।

(iii) Basic File System: যদি একটি ফাইল টেবিলে আগে থেকেই একটি ফাইল থোলা থাকে, সেকেতে VTOC Entry Number ব্যবহার করা হয় এবং যদি ফাইল থোলা না থাকে তাহলে একটি ফাইল টেবিলে কপি করা হয়।

(iv) File Organisation Strategy Module (FOSM): এ ধাপে Logical Byte Address টি Logical Number এ অফসেট (Offset) এ জোগান্ত হয়। ফাইল ম্যাপ (File Map) ব্যবহারের মাধ্যমে Logical Block নামান্তি Physical Block নামান্তে রূপান্তর হয়। যদি কাঞ্চিত Block টি Logical Block নামান্তে রেকর্ড সমূহ Caller কে ফেরত পাঠানো হয়, আর যদি কাঞ্চিত Block টি Block এ না থাকে, তবে Device Strategy Module টি Physical Block কে Call করে এবং Read করে। Read করার পর লজিকাল রেকর্ড সমূহের Caller কে ফেরত পাঠানো হয়। আবার যদি Write Request হয়, তবে একইভাবে Physical Block টি Allocation Strategy Module দ্বাৰা Assign করা হয়।

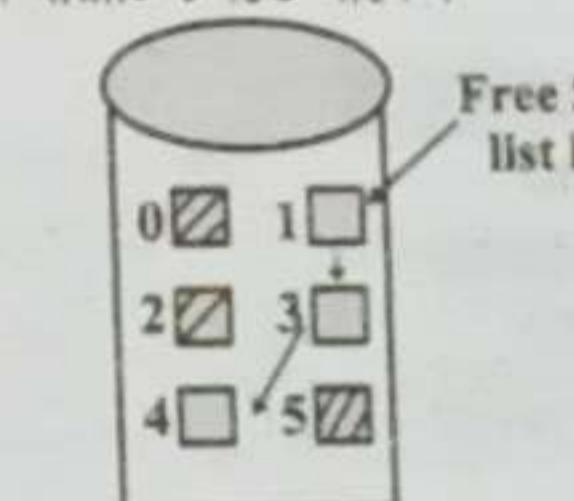
(v) Allocation System Strategy Module (ASM): এই ধাপে একটি Free (মুক্ত) Physical Block কে Free (মুক্ত) ফাইল ম্যাপে অপসারিত করা হয় এবং পরে তা মনোনীত ফাইলের ফাইল ম্যাপ(File Map) এ ছাপন করা হয়।

(vi) Device Strategy Module (DSM): এই ধাপে Physical Block নামান্ত এক্সেস ফরম্যাটে রূপান্তর হয়। এক্সেস ফরম্যাট বলতে ডিভাইস এর সিলিভার, নামান্ত, Track নামান্ত এবং লজিকাল রেকর্ড নামান্ত ইত্যাদিকে বুঝানো হয়েছে। এরপরে প্রযোজনীয় I/O Command ছাপন ও তারপর Control I/O সিস্টেমে প্রবেশ করে।

(vii) I/O Control System (IOCS): এই ধাপে I/O Channel ব্যবহার করা, Schedule করা, I/O Command কে Initialization করা ও Interrupt Procedure অনেক করা হয়। যখন I/O অপারেশন শেষ হয় তখন নিয়ন্ত্রণ Caller কে ফেরত পাঠায়।

প্রশ্ন-৮. FAT এবং NTFS এর মধ্যে পার্থক্য তালো লিখুন।

উত্তর: FAT এবং NTFS এর মধ্যে পার্থক্য তালো হলো:



(iii). Grouping: এ পক্ষতিতে প্রথম Free Block এন্ড সংস্ক্রিত Free Block এর Address জমা করে রাখে। এতে প্রথম (n-1) সংস্ক্রিত Block Actual Free থাকে। সবশেষ ব্লক টি অন্ত ন সংস্ক্রিত Free Block এর Address ধারণ করে।

(iv). Counting: এই পক্ষতিতে প্রস্তুত সংলগ্ন কতকগুলি Block Allocated কিংবা Free হয়। তাই আমাদের First Free Block Address এবং কতগুলি Block Free তা Count করে রাখতে হয়।

প্রশ্ন-১০. General (সাধারণ) ফাইল System এর বৈশিষ্ট্য বা Feature সমূহ লিখুন।

উত্তর: General (সাধারণ) ফাইল System এর বৈশিষ্ট্য সমূহ নিম্নরূপ:

- a) কম সংখ্যাক I/O অপারেশন।
- b) Logical Record & Physical Block সাথে ইত্যাজের মধ্যে Flexibility ব্যবহা।
- c) File ক্ষেত্রের ব্যঞ্জন Calculation.
- d) File Space এর Dynamic Calculation.
- e) সুবিধাজনক File নামের ব্যবহা।

প্রশ্ন-১১. সেকেন্ডারি মেমোরিকে কয় ভাগে ভাগ করা যায়? বর্ণনা দিখুন।  
উত্তর: সেকেন্ডারি মেমোরিকে পঠন ও কার্যতন্ত্রে নিখনিলিখিত ভাগে ভাগ করা যায়। ধরা:  
(a). Hard Disk: Hard Disk এক প্রকার সেকেন্ডারি মেমোরি। যাতে একই সাথে Read & Write করা যায়। ইহ এক প্রকার হার্ড মেমোরী যা User কর্তৃক খুব সহজে Delete বা Write করা যায়।

**Hard Disk এর অংশ সমূহ:**

1. Read Write Head
2. Sector
3. Plate
4. Track Surface
5. Drive Spindle
6. Boom

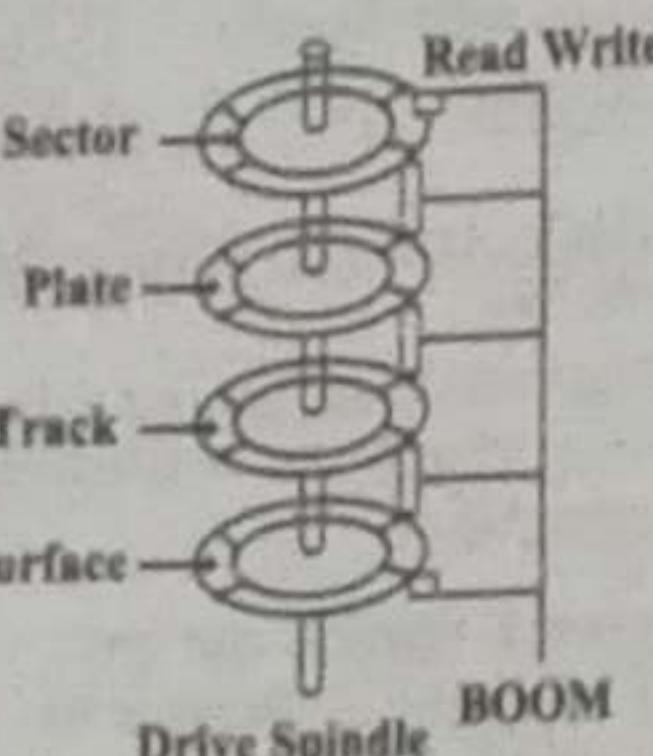


Figure: Hard Disk বিভিন্ন অংশ

(b). Floppy Disk: Floppy Disk এর অকার Secondary Memory যাতে Read এবং Write Operation এর মধ্যে তথ্য সংকেন করে।

**Floppy Disk এর অংশ সমূহ:**

1. Track
2. Inter Track Gap
3. Sector
4. Inter Sector Gap
5. Read Write Head
6. Moving Mechanism.

(c). CD (Compact Disk)

(d). DVD (Digital Versatile Disk)

প্রশ্ন-১২. CD ROM এর চির সহ বর্ণনা করন। লিখুন।

উত্তর: CD ROM হল Compact Disk Read Only Memory যাতে শুধুমাত্র একবার Write করা যায় এবং একাধিক বার Read করা যায়। তবে বর্তমানে একাধিক বার Read Write ক্ষমতা সম্পর্কে CD বাজারে পাওয়া যায়।

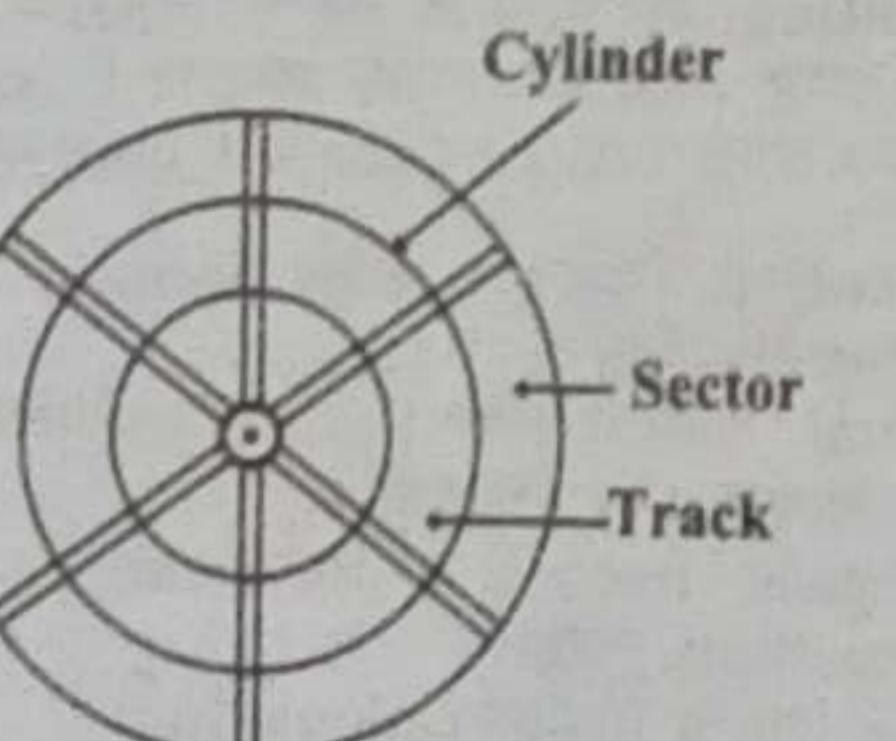


Figure: CD ROM বিভিন্ন অংশ

**CD এর পঠন অংশ সমূহ হল:**

- a) Cylinder: Cylinder প্রাণিক এর তৈরী। যার উপর Allumiunium এর প্লেপ দেওয়া থাকে।
- b) Sector: এটিটি সমকেন্দ্রীক বৃত্তের অংশকে Sector বলে।
- c) Track: সমকেন্দ্রীক বৃত্তগুলোকে Track বলে।

প্রশ্ন-১৩. Secondary Storage Management এর কাজ কি? উত্তর: Secondary Storage Management এর কাজ:

- a) Free Space Management
- b) Storage Allocation
- c) Disk Scheduling

প্রশ্ন-১৪. ফাইল সিস্টেম (File System) কি? বিভিন্ন প্রকার ফাইল সিস্টেম সম্পর্কে আলোচনা করন। [NTRCA (Lecturer)-2011]  
উত্তর: ফাইল সিস্টেম: তথ্যের লজিক্যাল Organization এর সাথে File System সম্পর্ক যুক্ত। একটি Computer System এ Physical Storage Device এ Data Store ও পুনরুৎসব ক্ষমতাকাজ সমূহ OS এর যে অংশ সম্পাদন করে তাকে File System বলে।

একটি Computer System এ বিভিন্ন প্রকার ফাইল সিস্টেম ব্যবহার করা যায়, যেখনে কিনা ধাপে ধাপে ডেভেলপ করা হয়েছে। যেমন:

- a) FAT
  - b) NTFS
  - c) HFS+
  - d) EXT4
  - e) ZFX ইত্যাদি।
- বিভিন্ন প্রকার ফাইল সিস্টেম সম্পর্কে আলোচনা:
- FAT (File Allocation Table):** কম্পিউটিং জগতের সবচাইতে পরিচিত এবং সবচাইতে বেশি ব্যবহৃত কম্পিউটার ফাইল সিস্টেম হয়ে ফ্যাট বা এফএটি (FAT) যার সম্পূর্ণ নাম হচ্ছে ফাইল আলোকেন টেবিল (File Allocation Table) এবং এটি মাইক্রোসফট (১৯৭৭ সালে) ডেভেলপকৃত একটি ফাইল সিস্টেম। এই ফাইল সিস্টেমটি আস অনেক দিন হয়ে গেল এবং আসার পরে এর কিছু আপডেটও হয়েছে। যেমন: FAT12, FAT16 এবং FAT32।
- যতোগোলো প্রধান ফাইল সিস্টেম রয়েছে এদের মধ্যে FAT সবচাইতে সহজ একটি সিস্টেম এবং পুরুষীর প্রায় সকল অপারেটিং সিস্টেমই এই ফাইল সিস্টেমকে সৌন্দর্য (Read) করার ক্ষমতা রাখে। এটি শুধু ফাইল সিস্টেমকে সৌন্দর্য করার জন্য একটি প্রতিক্রিয়া করে। তবে হার্ডড্রাইভের (Hard Drive) কোন ফাইল সম্বন্ধে প্রস্তুত হলে এই সিস্টেমটি তা ঠিক করার ক্ষমতা রাখে না। ফ্যাটের সবচাইতে বড় অনুবিধি হচ্ছে এই ফরম্যাটটি শুধু 8 জিবি (4 GB) পর্যন্ত এক একটি ফাইল সংরক্ষিত করতে পারে।
- একটি ফাইলের সাইজ 8 জিবির (4 GB) বেশি হলে ফ্যাট তা হ্যালে (Handle) করতে পারে না। আজকের দিনে হার্ডড্রাইভে (Hard Drive) আর এটি ফরম্যাট ব্যবহার করা যায় না তবে যেমনের কার্ড (Memory Card) বা ফ্ল্যাশ ড্রাইভ/পেনড্রাইভ (Flash Drive/Pen Drive) উল্লেখ এই ফাইল সিস্টেম ফরম্যাট এখনো ব্যবহার করা যায়। যেকোনো ড্রাইভকে এই ফাইল সিস্টেমে ফরম্যাট করা একেবারেই সহজ। FAT32 বিশেষ করে পুরাতন ভার্সন উইন্ডোজ অপারেটিং সিস্টেম উল্লেখ করা হতো। যেমন: Windows 95, Windows 98 ইত্যাদি।
- Operating System
- NTFS (New Technology File System):** New Technology File System (NTFS) (নিউ টেকনোলজি ফাইল সিস্টেম বা এনটিএফএস) হলো মাইক্রোসফট (১৯৯৩ সালে) এর ডেভেলপকৃত আরেকটি কম্পিউটার ফাইল সিস্টেম; যা কিনা পরবর্তী ইন্ডোজ (Next Generation) উদ্দেশ্য করে তৈরি করা হয়েছে। এর পঠন প্রণালী অনেক জটিল এবং এটি শুধু মাইক্রোসফটের অপারেটিং সিস্টেমের জন্য ব্যবহৃত হয়। অর্থাৎ Windows এবং Windows XP Operating System এ এটি সর্বাঙ্গিন ব্যবহার করা হয়।
- Tera Byte) বেশি পর্যন্ত সমর্থন করতে পারে এবং এটি অবশ্যই একটি জার্নালিং (Journaling) ফাইল সিস্টেম, ফলে কোন তুটি (Error) দেখা দিলে তা ফিক্স (Fix) করতে অনেক সহজ হয়ে যায়। শুধুমাত্র উইন্ডোজ অপারেটিং সিস্টেম এই ফাইল সিস্টেম ফরম্যাটকে ব্যবহার করতে পারে না, তবে Mac OS এবং Linux একে ব্যবহার করার ক্ষমতা রয়েছে।
- EXT4:** EXT4 বর্তমানে Linux নির্ভর অপারেটিং সিস্টেমে বহুব্যবহৃত একটি ফাইল সিস্টেম। এটি শুধুমাত্র ভার্সন EXT2 এবং EXT3 এর সফল ভার্সন। যেখানে কিছু জটিল টেকনিক ব্যবহার করে এর পারফরমেন্সকে (Performance) উন্নতি করা হয়েছে। এটি শুধুমাত্র জার্নালিং (Journaling) ফাইল সিস্টেম। Removeable Media Storage এ এটিকে ব্যবহার করা যায়, Removeable Media Storage Corrupted (বিকৃত) হয়ে যেতে পারে। তবে অবশ্যই এটি হার্ডড্রাইভের (Hard Drive) জন্য প্রযোজ্য। এটি শুধুমাত্র একটি ফাইল সাইজ ১৬ টেরাবাইট (16 Tera Byte) এবং স্টোরেজ সাইজ ১ এক্সাবাইট (1 Exa Byte) পর্যন্ত সাপোর্ট করে। এটি শুধুমাত্র অপারেটিং সিস্টেমের জন্য নয়, তবে EXT4 Read নামক একটি প্রযোগ ব্যবহার করে উইন্ডোজ ইউজর্স (User) অস্তিত্বে EXT4 ফাইল সিস্টেম ফরম্যাটকে সীড (Read) করতে পারে।
- ZFX:** ZFX শুধু একটারপ্রাইজ সেভেলের কম্পিউটার ফাইল সিস্টেম ফরম্যাট। এটি হোম (Home) থেকে বেশি একটারপ্রাইজ (Enterprise) কাজে ব্যবহার করা যায়। বিশেষ করে সার্ভার স্টোরেজে (Server Storage) একে ব্যবহার করা যায়, যেখানে একসাথে মিলিয়ন ইউজারের সরক্ষিত পার্টিশন স্টোরেজে সাইজ ১৬ এক্সাবাইট (16 Exa Byte) পর্যন্ত সাপোর্ট করে। এটি শুধুমাত্র ম্যাক সার্ভার (Mac Server), লিনাক্স (Linux) এবং হোট থাটো ইউনিক্স (Unix) অপারেটিং সিস্টেমে সাপোর্ট করে।
- প্রশ্ন-১. ফাইল সিস্টেম কি?**
- উত্তর: তথ্যের লজিক্যাল Organization এর সাথে File System সম্পর্ক যুক্ত। একটি Computer System এর Physical Storage Device এ Data Store ও পুনরুৎসব করা সংক্রান্ত কাজ সমূহ OS এর যে অংশ সম্পাদন করে তাকে File System বলে।
- প্রশ্ন-২. File Pointer কি?**
- উত্তর: File এর Read বা Write Operation এ ব্যবহৃত Pointer টিকে File Pointer বলে।
- প্রশ্ন-৩. File Open Count কি?**
- উত্তর: সর্বশেষ File টি Close না করা পর্যন্ত কতবার File Open এবং Close করা হয় তার সংখ্যা count করে।
- প্রশ্ন-৪. Disk Location of File কি?**
- উত্তর: Disk Location of File ব্যবহার করে Disk এর কোন Location এ File আছে তা Memory তে Store করা যায়।
- প্রশ্ন-৫. Text File কি?**
- উত্তর: Character Organized হয়ে Line বা Page আকারে কোন File তৈরী করা হলে তাকে Text File বলে।
- প্রশ্ন-৬ :** Suppose we have a 100 RPM disk with 8 heads and 480 cylinders. It is divided into 120-cylinder zones with the cylinder in different zones containing 200, 240 (Maximum Storage) হিসেবে ১০ লাখ টেরাবাইটের (10M
- bitBox ICT Master Copy - 712
- bitBox ICT Master Copy - 713

and 320 sectors. Assume each sector contains 4096 bytes and seek time between adjacent cylinders of 2 ms. [Ans: ANS-2021-AP]

a. What is the disk capacity?

b. What is the maximum transfer rate?

**উত্তর:** To get the capacity of the disk, multiply the total number of sectors by the sector size.

The total number of sectors:

$$120*200 + 120*240 + 120*320 = 91200$$

so, the total disk capacity:  $91200 * 4096 = 373555200$  bytes. = 356 MB. (Ans)

To get the transfer rate, first, pick the cylinder zone with the largest number of sectors (320). This is the cylinder zone that will yield the greatest transfer rate. It's probably near the outside of the disk (since there's more surface area to work with).

Now, multiply the number of sectors by the number of bytes per sector ( $320 * 4096$ ) and you get 1310720. This is the number of bytes you can retrieve per disk rotation.

The number of rotations per second is  $100/60$ , approximately 1.67. Multiply that by 1310720 bytes per rotation, and you get 2188902, or about 2.08 megabytes per second. (Ans)

Since you need 1.67 cylinders to get a full second's worth of data, you should subtract the amount of data that cannot be read while the head is seeking between cylinders (2 ms per seek).

**প্র-৭:** Consider a disk pack with the following specifications- 16 surfaces, 128 tracks per surface, 256 sectors per track and 512 bytes per sector.

Answer the following questions-

i. What is the capacity of disk pack?

ii. If the format overhead is 32 bytes per sector, what is the formatted disk space?

iii. If the disk is rotating at 3600 RPM, what is the data transfer rate? [Rupali bank -ANE-2021]

**Ans:** i. Capacity of disk pack = Total number of surfaces  $\times$  Number of tracks per surface  $\times$  Number of sectors per track  $\times$  Number of bytes per sector

$$\begin{aligned} &= 16 \times 128 \times 256 \times 512 \text{ bytes} \\ &= 228 \text{ bytes} \\ &= 256 \text{ MB} \end{aligned}$$

ii. Total number of sectors = Total number of surfaces  $\times$  Number of tracks per surface  $\times$  Number of sectors per track

$$\begin{aligned} &= 16 \times 128 \times 256 \text{ sectors} \\ &= 2^{19} \text{ sectors} \end{aligned}$$

Thus, Number of bits required to address the sector = 19 bits

**Formatting overhead** = Total number of sectors  $\times$  overhead per sector

$$\begin{aligned} &= 2^{19} \times 32 \text{ bytes} \\ &= 2^{19} \times 25 \text{ bytes} \\ &= 224 \text{ bytes} \\ &= 16 \text{ MB} \end{aligned}$$

Now, Formatted disk space = Total disk space - Formatting overhead

$$\begin{aligned} &= 256 \text{ MB} - 16 \text{ MB} \\ &= 240 \text{ MB} \text{ (Ans)} \end{aligned}$$

### iii. Data Transfer Rate:

Number of rotations in one second

$$\begin{aligned} &= (3600 / 60) \text{ rotations/sec} \\ &= 60 \text{ rotations/sec} \end{aligned}$$

Now, Data transfer rate: = Number of heads  $\times$  Capacity of one track  $\times$  Number of rotations in one second

$$\begin{aligned} &= 16 \times (256 \times 512 \text{ bytes}) \times 60 \\ &= 24 \times 28 \times 29 \times 60 \text{ bytes/sec} \\ &= 60 \times 221 \text{ bytes/sec} \\ &= 120 \text{ MBps (Ans)} \end{aligned}$$

## Protection and Security

(Classification and Handling Techniques)

**প্র-১:** Protection Systems প্রয়ন্তে নৃন্যতম বিশেষাবিকরণ (Privilege) মীতি কিভাবে সহায়তা করে? (How does the principle of least privilege aid in the creation of protection systems?)

**উত্তর:** Least Privilege (নৃন্যতম বিশেষাবিকরণ) এর Principle (মীতি/তত্ত্ব) ব্যবহারকারীদেরকে তাদের টাকের (tasks) উপর প্রয়োগ করা জন্য যথেষ্ট সুবিধা অনুমতি দিয়ে থাকে। এই মীতির ফ্রেমওয়ার্ক প্রযোগ করা একটি সিস্টেমের এমন একটি প্রোপার্টি যারে যা একটি উপাদানের ফেইলিউর বা কল্পনামাইজ (Failure or Compromise) সিস্টেমটির সর্বনিম্ন ক্ষতি করে কেননা ফেইলিউর বা কল্পনামাইজ (Failure or Compromise) কম্পোনেন্ট ইহার নরমাল মেরু অপারেশনকে (Normal Mode Of Operation) সাপোর্ট করে। Least set of Privileges এর দরকার।

**প্র-২:** প্রিভিল প্রকার একসেস (Access) এর নাম লিখুন।

**উত্তর:** প্রিভিল প্রকার একসেস (Access) এর নাম তলো নিম্নরূপ:

- a) Read
- b) Write
- c) Execute
- d) Append (সংযোজন করা)
- e) Delete
- f) List

প্র-৩: Access lists এবং Capability lists এর মধ্যে মূল পার্শ্ব কী?

**উত্তর:** Access lists: একটি আরেস লিস্ট হচ্ছে এমন একটি লিস্ট যা প্রত্যেক অবজেক্টের জন্য একটি Non Empty Set সহকারে ভোর্মেইনস গঠন করে এসকল অবজেক্ট আরেস করার অধিকারের জন্য।

**Capability lists:** একটি ক্যাপাবিলিটি লিস্ট হচ্ছে অবজেক্টের একটি লিস্ট এবং অপারেশন প্রত্যেক ভোর্মেইনের এই সকল অবজেক্ট সমূহের জন্য অনুমোদিত।

ব্যবহারকারীদের ভোর্মেইন কম্পিউটার গেইন থেকে ব্যবহারকারীদের যোগাযোগ করার জন্য এখন সময়ের হিসাবে ব্যবহারকারীদের যোগাযোগ ক্ষেত্র হচ্ছে একটি Revocation (বালিদ্বত্ব/প্রত্যাহারক্ত) Process অবশ্যই ঘটে। এই Revocation হতে পারে immediate, selective (since the computer staff may access it at any hour), total, and temporary (since rights to access will be given back later in the day).

**প্র-৪:** Consider a computing environment where a unique number is associated with each process and each object in the system. Suppose that we allow a process with number n to access an object with number m only if  $n > m$ . What type of protection structure do we have?

**উত্তর:** Hierarchical structure.

**প্র-৫:** কেন এমন একটি সিস্টেমকে রক্ষা করা কঠিন হেখানে ব্যবহারকারী তাদের নিজের আই/ও (I/O) করার অনুমতি পায়? (Why is it difficult to protect a system in which users are allowed to do their own I/O?)

**উত্তর:** আমরা জানি যে, কার্নেল এবং ইউজার মোড এর মধ্যে একটি পার্শ্ব হচ্ছে- কার্নেল মোড বিশেষাবিকরণ আপারেশন (যেমন: I/O) চালানোর জন্য ব্যবহৃত হয়। কেন I/O অবশ্যই কার্নেল মোডে পারফর্ম করতে পারে তার একটা কারণ হচ্ছে হার্ডওয়্যার আরেস করার জন্য I/O অবশ্যিক এবং System Integrity (সম্পূর্ণতা/বিতর্কতা) এর জন্য হার্ডওয়্যারের সঠিক আর্কেন্স প্রয়োজন। যদি ব্যবহারকারীদেরকে তাদের নিজের I/O তে পারফর্ম করার অনুমতি দেওয়া হয়, তাহলে System Integrity (সম্পূর্ণতা/বিতর্কতা) এর নিচয়তা নিতে পারবেন।

**প্র-১০:** কার্যকর ক্ষমতা ম্যানিপুলেশন (Efficient Capability Manipulation) এর জন্য একটি কম্পিউটার সিস্টেমে কি ধরণের বৈশিষ্ট্য সম্পর্ক হার্ডওয়্যারের প্রয়োজন হয়? এই ধরণের বৈশিষ্ট্য কি মেমোরি প্রোটেকশন এর জন্য ব্যবহার করা যাবে?

**উত্তর:** কার্যকর ক্ষমতা ম্যানিপুলেশনের (Efficient Capability Manipulation) জন্য একটি হার্ডওয়্যার বৈশিষ্ট্য যা কিনা একটি ক্যাপাবিলিটি অবজেক্টকে (a capability object) একটি আরেসিবল অবজেক্ট (a accessible object) বা উভয়ের প্রত্যেককে আইডেন্টিফাই করার জন্য অনুমতি দিয়ে থাকে। ইহা কাটিন মেমোরি প্রোটেকশনের জন্য ব্যবহার করা যাবে কারণ তারা একটি বাইনারী মান থেকে আলাদা সামান্য প্রোটেকশন এদান করে যা তারা একটি ক্যাপাবিলিটি অবজেক্ট বা অন্যকিছু কিনা তা নির্দেশ করে। মেমোরি প্রোটেকশনের জন্য ভার্চুয়াল মেমোরির বৈশিষ্ট্যের সম্পূর্ণ সাপোর্ট দরকার।

**Security:**

**প্র-৭:** একটি সিস্টেম বিবেচনা করুন, যেখানে শুধুমাত্র স্টুডেন্টসদের মধ্যে "কম্পিউটার গেইন" খেলা হয় 10 P.M. এবং 6 A.M., যাকাটি মেধাবৃত্ত মধ্যে 5 P.M. এবং 8 A.M. পর্যন্ত এবং কম্পিউটার সেটারের স্টাফদের বারা সবসময় খেলা হয়। নক্ষত্র সাথে এই মীতি ব্যবহারকারীদের জন্য একটি পক্ষতি সাজেক্ষন করুন।

**উত্তর:** একটি ডাইনামিক প্রোটেকশন স্ট্রাকচার তৈরি করলে তিনি ক্যাটাগরির ব্যবহারকারীদের জন্য সময় ব্রাচ অনুসৰে ব্যবহারযোগ্য রিসোর্স তলোর সেটকে পরিবর্তন করে দিবে। সময়ের পরিবর্তন হিসাবে,

উভয় কোম্প্যাক্টেড মিডল-মাল-আর্টিকের অবস্থা থেকে রক্ষা পেতে একটি প্রেরণ ও একটি আপক যোগাযোগ করা করার পূর্বে এর সেশন কী এর সরবরাহের জন্য সহজ হয়। উন্নতরূপ বক্সে, যদি কেট একটি কমন সেশন কী পরিচয় দেওয়ার জন্য সৃষ্টি যোগাযোগকারী মেশিনে একটি সিকিউর শেল প্রোটোকল ব্যবহার করতে চাই এবং যদি সেশন কী টি এক্সচেঞ্জ করার জন্য প্রোটোকল মেসেজটি ব্যবহার অব্যুক্তিক্রম মেরুনিজম রাখা সুরক্ষিত না থাকে, তখন আর্টিকের জন্য আপনা একটি সেশন কী তৈরী করা সহজ হয়ে যায় এবং বিনামূল সৃষ্টি যোগাযোগকারী ডিভাইসে ভেটা একসেস করার সুবিধা পেতে যায়।

বিলের ক্ষেত্রে, সার্ভার যদি সেশন কী তৈরী করার অনুমতি দেয়, আর্টিকের সার্ভার থেকে সেশন কী পেতে পারে, এবং ক্লায়েন্টের সাথে এই সেক্ষেত্র সেশন কী দিয়েই যোগাযোগ করে, একেবারে আর্টিকের নিরাপত্তা এবং ক্লায়েন্টেরকে সুরক্ষিত করে এই ক্লায়া সেশন কী ব্যবহার করার জন্য। যদি আর্টিকের ক্লায়েন্ট থেকে ভেটা হিসিত করে, ইহা ভেটাকে তিনিটি করে, পুনরায় অসল সেশন কী সহকারে ভেটাকে সার্ভার থেকে এনক্রিপ্ট করে এবং সার্ভারে এনক্রিপ্টেড ভেটা প্রেরণ করে আর্টিকের উপরিক্রমে ক্লায়েন্ট অবশ্য সার্ভারকে প্রেরণ করা যাবার্দিত।

এই ধরনের আর্টিকে সার্ভার থেকে আসা মেসেজের অবেদনটিকে করে ডিফিটাল সিগনেচার ব্যবহার করে এডিয়ে ঢলা যায়। যদি সার্ভার সেশন কী রাখা যোগাযোগ করতে পারে এবং ইহা একটি মেসেজের অবিভেদিত মাছাই করতে পারবে যা কিনা বীকৃত অবস্থিতি প্রেরণকৃত একটি ডিফিটাল সিগনেচার রাখা সুরক্ষিত থাকে। তখন আর্টিকের সেশন কী কে নকল করতে পারেনা, এবং এই ফাল মিডল-মাল-আর্টিকে এডিয়ে ঢলা যায়।

প্রশ্ন-২: একটি পাসওয়ার্ড বিভিন্ন উপায়ে অন্যান্য ব্যবহারকারীর কাছে পরিষ্কৃত মনে হতে পারে। এই ধরনের Event (অ্যটো) সন্তু করার জন্য এমন কোম সহজ পদ্ধতি আছে কি? ব্যাখ্যা করুন।

উভয় ধরনই একজন ব্যবহারকারী স্লগ ইন করে, তখন সিস্টেমটি Last Time হিসেবে যে ব্যবহারকারী সিস্টেমটিতে Logged On হিসেবে।

প্রশ্ন-৩: কম্পিউটার সিস্টেমে এনক্রিপ্টেড ভেটা সঞ্চয় করার সৃষ্টি সুবিধা প্রদান।

উভয় এনক্রিপ্টেড ভেটা অপারেটিং সিস্টেমস প্রোটোকলান ফ্যাসিলিটিজ রাখা সুরক্ষিত থাকে, সেই সাথে একটি পাসওয়ার্ড ও সরবরাহ তাদেরকে তিনিটি করার জন্য। কানুন ধর্ম নিরাপত্তার কথা আসে, তখন সৃষ্টি কী (Keys) অবশ্যই একটির ঢেরে তালো।

প্রশ্ন-৪: একটি ব্যাকেরের ক্ষেত্রে একটি কম্পিউটার সিস্টেমের জন্য নিরাপত্তা বিবেচনা করে একটি তালিকা তৈরি করুন। যেখানে আপনার তালিকার অন্তেক আইটেম কিনা শারীরিক নিরাপত্তা, মানবীয় নিরাপত্তা, অথবা অপারেটিং সিস্টেমের নিরাপত্তার সাথে সম্পর্কিত।

উভয় ধরণে একটি ব্যাকেরের ক্ষেত্রে একটি কম্পিউটার সিস্টেমের জন্য ৬ ধরনের নিরাপত্তা বিবেচনা করে একটি তালিকা তৈরি করা হলো।

- In a protected location, well guarded: physical, human.
- Network tamper proof: physical, human, operating system.
- Modem access eliminated or limited: physical, human.
- Unauthorized data transfers prevented or logged: human, operating system.
- Backup media protected and guarded: physical, human.

f) Programmers, data entry personnel, trust worthy: human

প্রশ্ন-৫: বাকর-ভেরেফেল আর্টিক (Buffer-overflow attacks) একটি তালো প্রোগ্রাম মেমোরোলজি অবলম্বন করে অথবা বিপ্লব হার্ডওয়ার সাপ্লেট ব্যবহার করে এডানো যেতে পারে। - এই সমস্যা অসুচিতা করুন।

উত্তর: One form of hardware support that guarantees that a buffer overflow attack does not take place is to prevent (প্রতিরোধ) the execution of code that is located in the stack segment of a process's address space. Recall that buffer-overflow attacks are performed by overflowing the buffer on a stack frame, overwriting the return address of the function, thereby jumping to another portion (অন্য অংশ) of the stack frame that contains malicious executable code, which had been placed there as a result of the buffer overflow.

By preventing the execution of code from the stack segment, this problem is eliminated (দূর). Approaches that use a better programming methodology are typically built around the use of bounds-checking to guard against buffer overflows. Buffer overflows do not occur in languages like Java where every array access is guaranteed to be within bounds through a software check. Such approaches require no hardware support but result in runtime costs associated with performing bounds-checking.

প্রশ্ন-৬: সকল পাসওয়ার্ডের তালিকা অপারেটিং সিস্টেমের মধ্যে হয় হয়। সুতরাং, কেন ব্যবহারকারী যদি এই তালিকা পত্র ব্যবহার করতে পারে, পাসওয়ার্ড প্রেটেকশন আর সরবরাহ করা হয়েন। একটি প্রাচী সার্জেন্ট কর্তৃ যা কিনা এই সমস্যা এডানো পারে। (Hint: বিভিন্ন ইন্টারনেল ও এল্টার্নেল প্রিঙ্গেজেন্টেশন ব্যবহার করুন।)

উভয় ইন্টারনেল পাসওয়ার্ডকে এনক্রিপ্ট করা হয় যাতে করে তাৰ পুনৰুৎকোচেত ঘৰ্মে একসেস হতে পারে। ত্ৰুমাত্র একজন ব্যক্তি ইহা একসেস করতে পারবে যা সিস্টেম অপারেট ইহা ডিকেন্ট কৰিব জন্য আৰু ধৰাকা উচিত।

প্রশ্ন-৭: এমন একটি সিস্টেম বিবেচনা করুন যা কিনা প্রতিদিন 10 মিলিঅক্ষিটি রেকৰ্ড জেনেরেট করতে পারে। ধৰে নিম, গড়ে সিস্টেমটি প্রতিদিন 20 টি করে আর্টিক করে এবং একটি আর্টিক 20 টি রেকৰ্ড এর মধ্যে রিস্ট্রাই (Reflected) হয়। যদি Intrusion-Detection System এর এলী true-alarm rate 0.6 এবং একটি false-alarm rate 0.0002 থাকে, Real Intrusions অনুমান সিস্টেমটি কত শতাংশ এক্সৰ্ব কেবলো করতে পারবে?

সমাধান: The probability of occurrence of intrusive records is  $= 10 \times (20/10^6) = 0.0002$

Using Bayes theorem,  
The probability that an alarm corresponds to a real intrusion is simply

$$= (0.0002 \times 0.6) / (0.0002 \times 0.6 + 0.9998 \times 0.0005) \\ = 0.1935796096$$

[NOTE: Security সম্পর্কে বিজ্ঞান Network Security Part 4 থেকে যাবে]

## ॥ Linux Commands ॥