



OPERATING SYSTEM APPLICATION

অনুশীলনী-১

অতিসংক্ষিপ্ত প্রশ্ন

১। অপারেটিং সিস্টেম কী?

উত্তরঃ- Operating System হলো এক ধরনের Program, যা Computer Hardware এবং User-এর মধ্যে সমন্বয় সাধন করে থাকে।

অন্যকথায়, Operating System হলো এক ধরনের বা একগুচ্ছ প্রোগ্রামের সমষ্টি যা, Computer-এর সকল Operation সম্পাদন করে এবং নিয়ন্ত্রণ করে।

২। কিছু Operating System-এর নাম লেখ।

উত্তরঃ- DOS, Windows, Mac OS, Unix, Linux, Xenix, Nobel, System-7, OS-2 ইত্যাদি।

৩। GUI-এর পূর্ণরূপ কী?

উত্তরঃ- GUI-এর পুরো নাম Graphical User Interface। GUI-কে Goo-ee উচ্চারণ করা হয়। Operating System -এর সাথে Interfacing-এর জন্য ব্যবহারকারী সরাসরি কমান্ড লাইন ইন্টারফেস ব্যবহার না করে Graphical Interface ব্যবহার করে, তাকেই GUI বা Graphical User Interface বলে। যেমন- System-7, Windows 95, 98, ME, XP, NT, 2000 etc. windows CE ইত্যাদি।

৪। কার্নেল কী?

উত্তরঃ- Operating System এর গুরুত্বপূর্ণ অংশ হলো কার্নেল। কার্নেল বিভিন্ন প্রোগ্রামসমূহকে একেকটি একেক সময়ে চলতে দেয়, অর্থাৎ Kernel প্রসেসর ব্যবস্থাপনায় প্রত্যেক Application-কে সময় ভাগ করে দেয়। কার্নেল অনেকগুলো প্রোগ্রামের মধ্যে সমন্বয় সাধন করে থাকে।

৫। Post কী?

উত্তরঃ- Post -এর পূর্ণ নাম হলো Power On Self Test. Rom এর অবস্থানকারী প্রোগ্রাম Post, যা Computer এর power switch কে on করার পর কার্যকরী হয় এবং Computer-এ সংযুক্ত বিভিন্ন Device সমূহের সেটিং চেক করে দেখে।

৬। Monitor Program কী?

উত্তরঃ- Operating System-এর যে অংশ ROM-এ অবস্থান করে Computer-এর যাবতীয় Operation সম্পন্ন হওয়ার জন্য নির্ধারক ও নির্ণয়কারী প্রোগ্রাম হিসেবে ব্যবহৃত হয়, তা হলো Monitor Program। একে Supervisor Program-ও বলে।

৭। Operating System এর ৪টি কাজ লেখ।

উত্তরঃ- নিম্নে Operating System -এর চারটি কাজ দেওয়া হলোঃ

- ১। User এবং Computer Hardware-এর মধ্যে সমন্বয় তৈরি করা।
- ২। মেমরি ম্যানেজমেন্ট করা।
- ৩। ফাইল ম্যানেজমেন্ট করা।
- ৪। CPU-কে বিভিন্ন টাস্ক এর জন্য assign করা ইত্যাদি।

৮। Multitasking-এর সুবিধা কী?

উত্তরঃ- একটি নির্দিষ্ট সময়ে একাধিক Task বা Program-কে কোনো Computer-এ execute করা যায়।

৯। ডস কী?

উত্তরঃ- DOS এর পূর্ণঅর্থ Disk Operating System । একে বর্ণভিত্তিক (Test Based) অপারেটিং সিস্টেমও বলা হয় । সাধারণত এতে কী-বোর্ডের সাহায্যে বিভিন্ন কমান্ড টাইপ করে কম্পিউটারকে নির্দেশ দেয়া হয় এবং কমান্ড অনুযায়ী কম্পিউটার কাজ করে ।

সংক্ষিপ্ত প্রশ্ন

১। অপারেটিং সিস্টেম এবং উদ্দেশ্য কয়টি ও কী কী?

উত্তরঃ- অপারেটিং সিস্টেমের দুই ধরনের Goal বা উদ্দেশ্য রয়েছেঃ

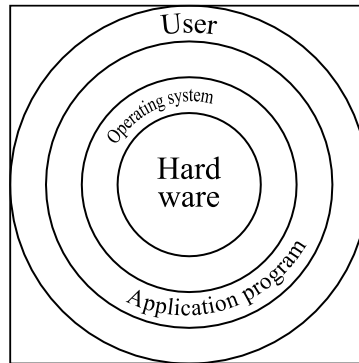
i) প্রধান উদ্দেশ্য (Primary Goal) :

- ব্যবহারকারীর Program কে Execute করা এবং User-দের Problem-গুলো সহজে সমাধান করা (Execute user programs and make solving user problems easier)
- কম্পিউটার সিস্টেমকে সুবিধাজনকভাবে ব্যবহার করা (Make the computer system convenient to user)

ii) কম্পিউটার Hardware-কে দক্ষতার সাথে ব্যবহার করা (Use the computer hardware in an efficient manner)

২। অপারেটিং সিস্টেম User কে কীভাবে Hardware থেকে আলাদা করে? চিত্র এঁকে ব্যাখ্যা কর ।

উত্তরঃ- সাধারণত একজন ব্যবহারকারী Operating System কে Application Program ও Data সরবরাহ করে এবং O/S হার্ডওয়্যারকে দিয়ে ব্যবহারকারীর প্রোগ্রাম নির্বাহ করে ফলাফল প্রদান করে । User মনে করে প্রদত্ত Command-এর কারণে Hardware নির্ধারিত কাজটি সম্পন্ন করেছে । কিন্তু প্রকৃতপক্ষে Operating System কীভাবে Hardware-কে দিয়ে কাজ করিয়ে নেয়, তা ব্যবহারকারীর নিকট উহ্য বা অস্পষ্ট থাকে । অর্থাৎ ব্যবহারকারী (user) ও Hardware -এর মধ্যে Operating System স্বচ্ছ পর্দার ন্যায় কাজ করে থাকে ।



৩। অপারেটিং সিস্টেমের কাজগুলো লেখ ।

উত্তরঃ- অপারেটিং সিস্টেম ব্যতীত একটি কম্পিউটার সিস্টেম অচল । অপারেটিং সিস্টেম দ্বারা কম্পিউটার তার যাবতীয় কার্যাবলি সাধন করে থাকে ।

- | | |
|--------------------------------------|--|
| i) প্রসেস ম্যানেজমেন্ট । | ii) মেইন মেমরি ম্যানেজমেন্ট । |
| iii) ফাইল ম্যানেজমেন্ট । | iv) I/O ম্যানেজমেন্ট । |
| v) সেকেন্ডারি স্টোরেজ ম্যানেজমেন্ট । | vi) কমিউনিকেশন ম্যানেজমেন্ট । |
| vii) ইনফরমেশন ম্যানেজমেন্ট । | viii) Hardware-এর অবস্থান পর্যবেক্ষণ করা । |
| ix) GUI ব্যবস্থাপনা করা । | x) সিপিইউ-কে বিভিন্ন টাস্কের জন্য অ্যাসাইন করা ইত্যাদি । |

৪। মনিটর প্রোগ্রাম ও অপারেটিং সিস্টেমের সম্পর্ক কী?

উত্তরঃ- Operating System বিভিন্ন ধরনের কাজ করে থাকে। প্রত্যেক কাজের জন্য এক বা একাধিক Program-এর প্রয়োজন হয়। এ সকল Program-কে একত্রে Operating System বলে।

Operating System-এর বিভিন্ন প্রোগ্রামের মধ্যে একটি নিয়ন্ত্রণকারী Program (Control Program), Main Memory-তে অবস্থান করে, যাকে Monitor Program বা Resident Program বা Supervisor Program বলে।

Operating System-এর Monitor Program ব্যতীত অন্যান্য Program-গুলো Secondary Memory-তে থাকে। এ সকল Program গুলোকে Transient Program বলা হয়। Utility Program, Assembler, Compiler ইত্যাদি এ অংশের অন্তর্গত। যখন Transient Program-এর দ্বারা কোনো কাজ করানোর প্রয়োজন পড়ে, তখন Monitor Program ঐ সকল Program-কে Main Memory-তে Load করে। কাজ শেষে ঐ সকল Program Main Memory থেকে মুছে (Delete) যায়। অর্থাৎ Secondary Memory-তে অবস্থিত Utility Program-কে Main Memory-তে সময়মতো, প্রয়োজনমতো Load করানোই হচ্ছে Monitor Program-এর কাজ।

৫। মনিটর প্রোগ্রাম কী? এটির কাজ লেখ।

উত্তরঃ- অপারেটিং সিস্টেমের বিভিন্ন অংশের মধ্যে Monitor Program হলো একটি গুরুত্বপূর্ণ অংশ। এটি Supervisor Program নামেও পরিচিত। এটি একটি Control Program। অপারেটিং সিস্টেমের যে অংশ রম (ROM)-এ অবস্থান করে Computer-এর যাবতীয় Operation execution হওয়ার জন্য নির্ধারক ও নির্ণয়কারী Program হিসেবে ব্যবহৃত হয়, তা হলো Monitor Program। একটি Microcomputer-এ Monitor Program নিম্নবর্ণিত কাজগুলো সম্পন্ন করে থাকে, যথাঃ

- i) POST (Power On Self-Test) অপারেশন Execute করা।
- ii) Computer, Keyboard, Interrupt, RAM, I/O port, Display প্রভৃতি অংশকে Initialize করা।
- iii) Computer-এর Boot routine সম্পন্ন করা।

৬। মাল্টিপ্রসেসর সিস্টেমের সুবিধা লেখ।

উত্তরঃ- Multiprocessor System-এর তিনটি প্রধান সুবিধা রয়েছে, যথা-

ক) Increased throughput: প্রসেসরের সংখ্যা বৃদ্ধি করে আমরা কম সময়ে ও সঠিকভাবে অধিক কাজ করতে পারি।

খ) Economic of scale: Multiprocessor system-এর খরচ Multiple single processor-এর খরচের তুলনায় কম। কারণ তারা Peripherals, mass storage ও power supply share করতে পারে।

গ) Increased reliability: যদি প্রত্যেক প্রসেসরের কাজ সঠিকভাবে ডিস্ট্রিবিউট করা যায় তাহলে একটি প্রসেসর অকেজো হলে পুরো system অকেজো হয়ে পড়বে না, বরং slow হয়ে পড়বে। কারণ বাকি প্রসেসরগুলো উক্ত প্রসেসরের কাজ ভাগাভাগি করে নেবে।

৭। Kernel বলতে কী বুঝ? কার্নেলের কাজ কী?

উত্তরঃ- কার্নেল হলো Operating System-এর Nucleus বা Operating System -এর heart। নিম্নে kernel-এর কাজ বর্ণনা করা হলোঃ

i) Kernel প্রোগ্রামসমূহ Main memory-তে load করে অর্থাৎ kernel-এর প্রয়োজনে দরকারি ট্রানজিয়েন্ট প্রোগ্রামকে Main memory-তে নিয়ে আসে এবং কাজ শেষে পুনরায় Secondary memory-তে ফিরিয়ে দেয়।

ii) Kernel বিভিন্ন প্রোগ্রামসমূহকে একেকটি একেক সময়ে চলতে দেয় অর্থাৎ kernel প্রসেসর ব্যবস্থাপনায় প্রত্যেক application-কে সময় ভাগ করে দেয়। এতে প্রসেসর time slice অনুযায়ী Multiuser ও Multitasking পরিবেশে কাজ করতে পারে।

iii) Kernel অনেকগুলো প্রোগ্রামের মধ্যে সমন্বয় সাধন (Co-ordination) করে থাকে।

iv) Memory management-এর মাধ্যমে Operating System প্রত্যেক Job-এর জন্য স্থান নির্দিষ্ট করে দেয়। ফলে System ত্রুটি করার সম্ভাবনা কম থাকে।

v) Multiprogramming ব্যবহার সম্ভব হয় বলে System-এর throughput বৃদ্ধি পায়।

৮। Multitasking বলতে কী বুঝায়?

উত্তরঃ- একটি নির্দিষ্ট সময়ে একাধিক Task বা Program-কে কোনো Computer-এ execute হওয়ার ability-কে Multitasking বলে (The ability to execute more than one task at the same time, atask being a program)। Multitasking এবং Multiprocessing অনেক সময় একই অর্থে ব্যবহৃত হয়ে থাকে যদিও Multiprocessing-এর ক্ষেত্রে একাধিক CPU থাকে। Multitasking-এর ক্ষেত্রে শুধুমাত্র একটি CPU কাজ করে, কিন্তু এতে খুব দ্রুত এক Program থেকে অন্য Program-এ CPU Switching করে থাকে। ফলে মনে হয় সকল Program-ই একই সময়ে Execute হয়।

অন্যকথায়, যে পদ্ধতির মাধ্যমে একাধিক Program কম্পিউটারের Main Memory-তে অবস্থান করে CPU-এর পূর্ণ প্রসেসিং যোগ্যতার সদ্যবহার করতে পারে, তাকে Multitasking বলে।

৯। Unix-এর বৈশিষ্ট্য লেখ।

উত্তরঃ- Unix-এর বৈশিষ্ট্যঃ

i) Unix, High-level language C-তে লেখা বলে এটি Super, mainframe, mini ও microcomputer-এ ব্যবহার করা যায় এবং একটি হতে অপরটিতে সহজে রূপান্তর করা যায়।

ii) এটি User-দের সাথে সহজে Interface তৈরি করতে পারে। এটি খুবই Powerful বলে User-এর সকল চাহিদা পূরণে সক্ষম হয়।

iii) Unix, hierarchical file system ব্যবহার করে, যাতে সহজে ও দক্ষভাবে রক্ষণাবেক্ষণ ও বাস্তবায়ন করা সম্ভব।

iv) Unix, Peripheral device-কে File-এর ন্যায় মনে করে। ফলে সাধারণ File-এর ন্যায় Peripheral Device-কে File-এর নাম দিয়ে সহজে ঐ Program সমূহ Access করা সম্ভব।

v) একসঙ্গে একাধিক ব্যবহারকারী Unix operating system O/S-এ কাজ করতে পারে।

১০। Linux-এর বৈশিষ্ট্য লেখ।

উত্তরঃ- Linux-এর বৈশিষ্ট্যঃ

i) Linux হলো Fast, free এবং Stable operating system.

ii) এটি -Unix এর সাথে অতি সহজেই Compatible.

iii) Linux-এর Source code বিনামূল্যে পাওয়া যায় বলে Unix workstation-এর বিকল্প হিসেবে ব্যবহৃত হচ্ছে।

iv) বর্তমানে Linux-এর Version টি FSF (Free Software Foundation)-এর Product.

v) Linux-এর জনপ্রিয়তার অনেক কারণের মধ্যে বিনামূল্য ও সহজলভ্যতা অন্যতম।

১১। বিভিন্ন ধরনের -এর নাম লেখ।

উত্তরঃ- নিম্নে বিভিন্ন ধরনের -এর নাম দেয়া হলোঃ

- i) সিরিয়াল ব্যাচ প্রসেসিং (Serial batch processing)
- ii) সরল মাল্টিপ্রোগ্রামিং (Simple multiprogramming)
- iii) মাল্টিপ্রসেসিং সিস্টেম (Multiprocessing system)
- iv) রিয়াল টাইম সিস্টেম (Real time system)
- v) টাইম শেয়ারিং সিস্টেম (Time sharing system)
- vi) অনলাইন অপারেটিং সিস্টেম (Online or Interactive mode O/S)
- vii) ডেডিকেটেড অপারেটিং সিস্টেম (Dedicated operating system)
- viii) Multitasking Operating System
- ix) Multithreading Operating System
- x) Distributed Operating System
- xi) Micro-kernel Based Operating System
- xii) Virtual Storage Based Operating System.

রচনামূলক প্রশ্ন

১। অপারেটিং সিস্টেমের কার্যাবলি বর্ণনা কর।

উত্তরঃ- অপারেটিং সিস্টেমের কার্যাবলি (Functions of operating system) :

Computer Hardware এবং Computer user-এর মধ্যে সমন্বয় (Interface) তৈরি করা হচ্ছে Operating system (O/S)-এর প্রধান কাজ। O/S ছাড়া কম্পিউটার চালানো অসম্ভব। নিম্নে O/S-এর বিভিন্ন রকম কাজ বর্ণনা করা হলোঃ

i) প্রসেস ম্যানেজমেন্ট (Process Management), অপারেটিং সিস্টেম (Operating system)-এর জন্য নিম্নলিখিত কাজগুলো করে থাকেঃ

- User process তৈরি (create) এবং delete করা
- System process তৈরি করা (create) এবং মুছে (delete) ফেলা
- Process suspension এবং resumption করা
- Process synchronize করা
- Process communicate করা
- Deadlock handling করা।

ii) মেইন মেমোরি ম্যানেজমেন্ট (Main memory management) :

- প্রধান মেমোরি ও অন্যান্য Storage-কে system program এবং ডাটা অংশে Allocate করে।
- Memory-র কোন অংশ ব্যবহৃত হচ্ছে, তার হিসেব রাখে।
- যে Process-গুলো memory-তে load হচ্ছে এবং কখন release হচ্ছে, তা Manage করা।
- প্রয়োজন অনুসারে Memory allocate এবং Deallocate করা।

iii) ফাইল ম্যানেজমেন্ট (File Management) :

- কোনো File-কে create এবং delete করা।
- Dictionary-গুলোকে create এবং delete করা।
- File এবং Dictionary-কে Manipulate করার জন্য Primitives support দেয়া।
- Secondary storage-এ File-কে Mapping করা।
- Backup file তৈরি করা।

iv) I/O ম্যানেজমেন্ট (I/O Management) :

- Buffering, caching এবং spooling.
- Input এবং Output এর সমন্বয় সাধন করা।
- I/O Device-এর মাধ্যমে একাধিক Program সমন্বয় ও নিয়ন্ত্রণ করা।

v) Secondary storage management:

- Free space management.
- Storage allocation
- Disk scheduling করা।

vi) Communication management:

- Create, delete communication connection.
- Send, receive messages.
- Transfer status information.
- Attach or detach remote devices.

vii) Information management:

- Managing groups of file.
- Managing file directories.
- Processing and managing the records within a file.

উপরিউক্ত প্রধান কাজ ছাড়া **Operating system** নিম্নলিখিত কার্যাবলি সম্পন্ন করে থাকেঃ

viii) Hardware-এর অবস্থান পর্যবেক্ষণ।

ix) Hardware-এর Error চিহ্নিতকরণ এবং সংশোধনের প্রয়োজনীয় ব্যবস্থা গ্রহণ।

x) Main memory-তে program লোড করা এবং পরিচালনা করা।

xi) Operator-এর নির্দেশ গ্রহণ, বিশ্লেষণ এবং সম্পাদন।

xii) নিরাপত্তা ও গোপনীয়তা রক্ষার জন্য নির্ধারিত Password ব্যবহারের সুবিধা প্রদান।

xiii) Sequential ভাবে batch processing-এর সুবিধা প্রদান।

xiv) Command ও Instruction-সমূহের translate করা।

xv) GUI ব্যবস্থা করা।

xvi) Internal (অভ্যন্তরীণ) টাইম ক্লক নিয়ন্ত্রণ।

xvii) Job transition-এর সুবিধা প্রদান।

xviii) Human-Computer এর মধ্যে Interaction প্রদান করা।

xix) Computer-কে Boot করা এবং Operation শুরু করা।

xx) Job scheduling করা।

xxi) Data এবং File storage-কে manage করা।

xxii) CPU-কে বিভিন্ন task-এর জন্য assign করা।

xxiii) অন্যান্য security এবং control-এর ব্যবস্থা করা।

xxiv) বিভিন্ন resource-কে allocate করা।

xxv) প্রোগ্রাম নির্বাহের সময় কোনো ত্রুটি থাকলে এটির message প্রদান ও debugging-এর সুবিধা প্রদান।

অপারেটিং সিস্টেমের প্রত্যক্ষ উপস্থিতি ছাড়া কম্পিউটারের কোনে operation-ই সম্পন্ন করা সম্ভব নয়।

অনুশীলনী-৩

অতিসংক্ষিপ্ত প্রশ্ন

১। Throughput কী?

উত্তরঃ- Operating system দ্বারা নির্দিষ্ট সময়ে মোট কাজের পরিমাণ।

২। JCL কী?

উত্তরঃ- JCL হচ্ছে Job Control Language, যা দ্বারা Batch system-এ একটি program হতে অন্য program-কে পৃথক করার জন্য ব্যবহার করা হয়।

অথবা, Batch processing system-এ এক প্রোগ্রাম থেকে অন্য প্রোগ্রামে Transit করার ক্ষেত্রে বা নতুন Device ব্যবহারের ক্ষেত্রে Operating system-এর জন্য কিছু নিয়ন্ত্রণকারী তথ্যের প্রয়োজন হয়। এসব নিয়ন্ত্রণকারী নির্দেশকে JCL বা Job Control System বলে। প্রত্যেক Program ও Data আগে ও পরে JCL নির্দেশ বসানো হয়।

৩। Spooling কী?

উত্তরঃ- বিভিন্ন Peripheral device-এর মধ্যে Data transmission-এর ক্ষেত্রে গতির যে অসামঞ্জস্যতা দেখা দেয়, তা দূর করে CPU-এর অলস সময়কে কমিয়ে প্রসেসিং-এর গতি বৃদ্ধি করার পদ্ধতিকে বলা হয় Spooling।

৪। JCL Computer কে কী কী তথ্য প্রদান করে?

উত্তরঃ- Name of Job (জবের নাম)

i) Name of user, Name of account no (ব্যবহারকারীর নাম ও হিসাব নং)

ii) Select I/O device (ব্যবহৃত I/O device)

iii) Select Assembler or Compiler (ব্যবহার্য (Compiler ও Assembler)

৫। ব্যাচ প্রসেসিং সিস্টেম বলতে কী বুঝ?

উত্তরঃ- Batch system-এর কতকগুলো Job সংগ্রহ করা হয়। বিভিন্ন বৈশিষ্ট্যের উপর ভিত্তি করে ব্যাচ তৈরি হয়। Batch processing operating system উক্ত ব্যাচ হতে JCL দ্বারা Program-গুলোকে Separate করে এবং প্রয়োজনীয় Compiler বা Assembler দ্বারা Execute করে সাথে সাথে Desired output তৈরি করে।

৬। স্পুলিং-এর প্রয়োজনীয়তা লেখ।

উত্তরঃ- বিভিন্ন Peripheral device-এর মধ্যে Data transmission-এর ক্ষেত্রে গতির যে অসামঞ্জস্যতা দেখা দেয়, তা দূর করে CPU-এর অলস সময়কে কমিয়ে প্রসেসিং এর গতি বৃদ্ধি করার জন্য স্পুলিং ব্যবহার করা হয়।

সংক্ষিপ্ত প্রশ্ন

১। Batch processing system-এর সুবিধা ও অসুবিধাগুলো কী কী?

উত্তরঃ- সুবিধা (Advantages):

i) যেহেতু Job to Job transition, automatic হয়, সেহেতু দুই Job-এর মধ্যবর্তী সময় কমে যায় এবং পূর্ববর্তী Manually system অপেক্ষা অনেক কম সময় লাগে।

ii) কোনো Job যখন Output-এ থাকে, তখন Computer অন্য একটি Job execute-এ ব্যস্ত থাকে আবার Input অন্য একটি Job-এর Input নিয়ে Busy থাকে। মোটের উপরে Computer-এর Idle সময় কমে যায়।

iii) Input দেয়ার পর Operator-এর Output পর্যন্ত আর কোনো কাজ থাকে না, ফলে সে সময় অন্য কাজে ব্যবহার করতে পারে।

অসুবিধা (Disadvantages):

i) Batch processing system-এ কিছু সময় ধরে Job সংগ্রহ করা হয় এবং Batch তৈরি করা হয়। ফলে উক্ত সময়ের মধ্যে Data হারানো সম্ভাবনা থাকে।

ii) অনেকগুলো Job একত্রে থাকে বলে Input দেয়ার পর নতুন করে Priority (Job-এর গুরুত্ব বুঝে আগে পরে Execute করার ব্যবস্থা) Setup করা যায় না। ফলে গুরুত্বপূর্ণ কোনো Program Complete হতে সময় বেশি লাগে।

iii) Batch monitoring করার জন্য অতিরিক্ত Algorithm (Procedure of solution) দরকার হয়।

iv) যেহেতু Group আকারে Solution তৈরি হয়, কাজেই Group-এর Job-এর কোনো সমস্যার জন্য অন্য সকল Job-কে অপেক্ষা করতে হয়। ফলে সকলের Output পেতে দেরি হয়।

২। স্পুলিং কী এবং কেন দরকার?

উত্তরঃ- Simultaneous peripheral output line-এর সংক্ষিপ্ত রূপ হলো Spooling। Spooling হলো এক ধরনের Process যার সাহায্যে Batch mode operating system-এর গতি আরও বৃদ্ধি করা যায়। তা ছাড়া এ পদ্ধতিতে CPU-এর সাথে I/O device ও Storage device-এর গতির Synchronization করে। সম্পূর্ণ Process-কে দ্রুতগতিসম্পন্ন করা হয়। ফলে CPU-এর Idle সময় কমে যায়। Spooling-এর উদ্দেশ্য হলো Program এবং Data-সমূহকে দ্রুতগতিসম্পন্ন করা ও CPU-এর ব্যবহারের জন্য I/O মাধ্যমকে প্রস্তুত রাখা। Card reader, Printer ইত্যাদির গতি CPU হতে অনেক কম। দ্রুতগতিসম্পন্ন Card reader মিনিটে ৮০ কলামের ১৫০০ কার্ড পড়তে পারে।

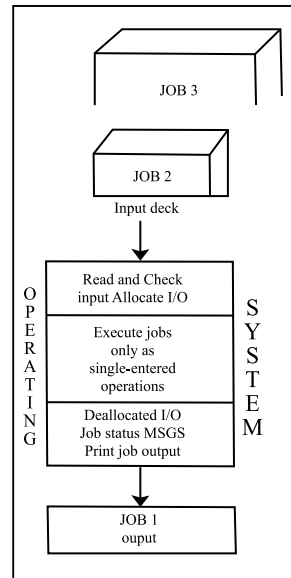
অর্থাৎ, প্রতি সেকেন্ডে মোট $(1500 \times 80 \times \frac{1}{60} = 2000)$ বাইট পড়তে পারে। তাহলে গতি হলো ২০০০ বাইট/সেকেন্ড। পক্ষান্তরে, Tape বা Disk-এর গতি হলো ১০০০০০ বাইট/সে.। তাহলে Tape বা Disk-এর সাথে Card reader এর গতির অসামঞ্জস্যতা এবং CPU-এর সাথে উভয়ের যে পার্থক্য রয়েছে, তা অপরিবর্তিত রেখে কাজের গতি বৃদ্ধিই হলো Spooling-এর প্রধান কাজ, যেখানে মাঝারি ধরনের Computer-এর গতি 3×10^6 বাইট/সে.।

রচনামূলক প্রশ্ন

১। Batch Processing System বর্ণনা কর।

উত্তরঃ- ব্যাচ প্রসেসিং পদ্ধতিতে একটির পর একটি ব্যবহারিক প্রোগ্রাম (Application Program) নির্বাহ করা হয়। প্রোগ্রাম নির্বাহের সময় কম্পিউটারের সামর্থ্য ও সুযোগ-সুবিধা শুধু ঐ প্রোগ্রামটির জন্য নিয়োজিত থাকে। ব্যাচ মোডে দীর্ঘ সময় ধরে ডাটা সংগ্রহ করে পরে একসাথে প্রয়োজনীয় প্রসেসিং-এর কাজ সম্পন্ন করা হয়। স্বয়ংক্রিয়ভাবে এক জব থেকে অন্য জবে যাওয়ার উপর ভিত্তি করে এ অপারেটিং সিস্টেম নির্মিত, যা সকল অপারেটিং সিস্টেমেই দেখা যায়। এ ব্যবস্থায় কম্পিউটার ব্যবহারকারীগণ তাদের প্রোগ্রাম ও ডাটা অনুযায়ী কার্ড পাঞ্চ করে। এ রকম বিভিন্ন জবের কার্ড একটার উপর আরেকটা সাজায় এবং এক সময় সব গৃহীত প্রোগ্রাম ও ডাটা কার্ডসমূহ প্রসেস করার জন্য কম্পিউটারে দেয়া হয়। কম্পিউটার একটার পর একটা নির্বাহ করে যায়। সব শেষে অপারেটর সকল জবের (যেভাবে ইনপুট করা হয়েছিল সেভাবে) প্রিন্টিং-এর কাজ সমাধা করে প্রয়োজ্য ব্যবহারকারীগণকে ফলাফল ফেরত দেয়। উল্লেখ্য যে, এ পদ্ধতিতে জব সংগ্রহ করে প্রসেসিং করতে দেয়ার সময় এবং প্রসেসিং শেষে ফলাফল সংগ্রহের সময় অপারেটরের প্রয়োজন হয়। এ ধরনের অপারেটিং সিস্টেমকে অনেক সময় সিরিয়াল, সিকুয়েন্শিয়াল, অফ-লাইন বা স্ট্যাক জব

প্রসেসিং সিস্টেমও বলা হয়। মাইক্রো কম্পিউটারে ব্যবহৃত CP/M এবং MS-DOS ব্যাচ মোড অপারেটিং সিস্টেমের উদাহরণ কম্পিউটার কার্ডে প্রোগ্রাম ও ডাটা সংগ্রহ করে ব্যাচ মোডে একসাথে নির্বাহ হওয়ার চিত্র এখানে দেখানো হলোঃ



চিত্রঃ ৩.১ সিরিয়াল ব্যাচ সিস্টেম অপারেশন

Batch processing-এ কয়েকটি Job একত্রে থাকে বলে Input, Input allocation. Execute one job, Deallocation of input, Job status change এবং Output same time-এ ঘটে। প্রত্যেক Job আলাদা আলাদা Function নিয়ে ব্যস্ত থাকে। কোনো Job stack-এ waiting কোনো Job executing এবং কোনো Job finished অবস্থায় থাকে। ফলে CPU idle সময় কমে যায়, যাকে Proper resource management বলা যায়।

ফলে পূর্ববর্তী System অপেক্ষা অনেকগুণ বেশি Efficient হয়। এ System-কে Stack system, Serial/sequential processing, Job mode বা Off line mode operating system বলা হয়।

ব্যাচ মনিটর অ্যালগরিদম নিম্নরূপঃ

- Step 1 : While jobs in the input queue.
- Step 2 : Do begin.
- Step 3 : Read program.
- Step 4 : Compile program.
- Step 5 : If there is no error
then execute object code
else give error message and quit.
- Step 6 : If error during execution
then
gives error message.
- Step 7 : End.

প্রত্যেকটি জব “Program begin” Statement এবং “Program end” Statement দ্বারা আলাদাভাবে শনাক্ত করা হতো। এ Statement-গুলো দ্বারা Batch monitor এক জব থেকে অন্য জবকে আলাদা করত। এ পদ্ধতি Automatic job to job transition-এর উপর ভিত্তি করে তৈরি করা হয়।

Microcomputer-এ ব্যবহৃত CP/M এবং MS-DOS Batch Mode operating system-এর উদাহরণ।

অনুশীলনী-৪

অতিসংক্ষিপ্ত প্রশ্ন

১। প্রসেস কী?

উত্তরঃ- নির্বাহিত কোনো Program-কেই Process বলা হয়। Program counter -এর Current value, register এবং variable-এর সমন্বয়ে process গঠিত হয়। এছাড়াও process-এর মধ্যে Process stack, temporary data যেমন- Subroutine parameters, return address, temporary ও global variable ইত্যাদি বিদ্যমান।

২। Process-এর state-গুলোর নাম লেখ।

উত্তরঃ- Process-এর state-গুলো নিম্নে দেয়া হলোঃ

(i) New, (ii) Running, (iii) Waiting, (iv) Ready, (v) Terminated.

সংক্ষিপ্ত প্রশ্ন

১। একটি প্রসেস এর অবস্থানগুলো বর্ণনা কর।

উত্তরঃ- একটি প্রসেস এর অবস্থানগুলো নিম্নে দেয়া হলো-

New: এ State-এ process-কে create করা হয়।

Running: যখন কোনো process-এর instruction -সমূহ processor কর্তৃক sequence অনুসারে সম্পাদন করবে।

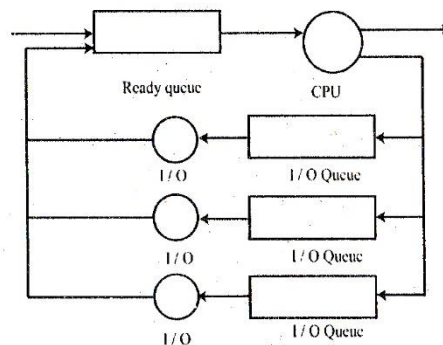
Waiting: এ State-এ process কিছু event সংঘটিত করার জন্য অপেক্ষা করে।

Ready: যখন Process running state-এ যাওয়ার জন্য সমস্ত শর্ত পূরণ করছে এবং processor-এর জন্য অপেক্ষা করছে। যখনই কোনো process blocked বা Ready অবস্থায় যায়, তখন সেই -কে সাময়িকভাবে থামানো বা বোলানো বা বলে।

Terminated: Process-এ State-এ তার execution শেষ করে।

২। সিডিউলিং কিউ বর্ণনা কর।

উত্তরঃ- Ready queue ব্যবহার হয় ঐ সমস্ত process-গুলো ধারণ করার জন্য, যেগুলো CPU পাওয়া জন্য ready আছে। অর্থাৎ, ঐ সকল process-গুলো, যারা ready এবং execute হওয়ার জন্য অপেক্ষা করছে, সে সকল process গুলোকে একটি List-এ রাখা হয়, সে List-কে Ready queue বলা হয়। এ List সাধারণত এক ধরনের Linked list। একটি Ready queue header-এর মধ্যে কোনো List-এর প্রথম এবং শেষ PCB-এর Pointers বিদ্যমান থাকে। প্রত্যেক PCB-এর একটি Pointer field থাকে, যা Ready queue-এর Next process-কে point করে।



চিত্রঃ Ready queue

Ready queue-য়ে First In First Out (FIFO) Queue হবে এমন কোনো কথা নেই। একটি Ready queue-কে FIFO Queue, priority queue, stack অথবা Unorder list একটি হিসেবে বাস্তবায়ন (Implement) করা যেতে পারে। সাধারণত Ready queue-র মধ্যে ঐ সকল process-গুলো অবস্থান করে, যারা CPU-তে Run করার জন্য অপেক্ষা করছে।

৩। Process এবং Program এর পার্থক্য লেখ।

উত্তরঃ- নিম্নে Process ও Program -এর মধ্যে পার্থক্য দেয়া হলোঃ

প্রসেস (Process)	প্রোগ্রাম (Program)
(i) মৌলিকভাবে নির্বাহ্য (Executable) কোন প্রোগ্রামই হচ্ছে প্রসেস।	(i) কতকগুলো Data এবং নির্দেশের সমন্বয় হচ্ছে প্রোগ্রাম।
(ii) Process একটি Active entity.	(ii) Program একটি Passive entity.
(iii) প্রসেস কতকগুলো State মেনে চলে।	(iii) প্রোগ্রাম শুধুমাত্র User-এর নির্দেশ মেনে চলে।
(iv) Program counter-এর Current value, Register এবং Valuable-এর সমন্বয়ে Process গঠিত হয়।	(iv) Disk-এ সংরক্ষিত file এবং তার উপাদান নিয়ে Program গঠিত হয়।
(v) প্রোগ্রামের মাধ্যমে দুটি Process-কে associate করা যায়। CPU-এর মাধ্যমে Process-সমূহের মধ্যে switching করা যায়।	(v) একটি Program নির্বাহের কাজ শেষ না হওয়া পর্যন্ত অন্য Program সম্পন্ন হয় না।
(vi) প্রসেস একটি চলমান কার্য।	(vi) Disk-এ সংরক্ষিত থাকে Program, এটি চলমান কার্য নয়।

৪। Race condition বলতে কী বুঝায়?

উত্তরঃ- যখন দু'টি Process-এর Scheduling এমন চরম সঙ্কটপূর্ণ (Critical) হয় যে, scheduling-এর বিভিন্ন নির্দেশের কারণে বিভিন্ন রকম ফলাফল প্রদান করে। দুই বা ততোধিক Process-এর প্রত্যক্ষ বা পরোক্ষ (Explicit or implicit) ডাটা শেয়ারিং বা রিসোর্স শেয়ারিং (Resource sharing)-এর ফলে Race condition পরিলক্ষিত হয়।

দুই বা ততোধিক Process যদি প্রায় একই সময়ে কোনো Shared device বা file-কে access করতে চায়, তখন কোন Process-টি আগে access পাবে, তা সম্পূর্ণরূপে নির্ভর করে Process-সমূহের অবস্থার উপর। অর্থাৎ, যে Process-টি আগে race করেছে, সেটি আগে access পাবে এবং অন্যটি ঐ সময়ে কোন Service পাবে না। Multiprogramming system-এর এরূপ Process নির্বাচন করার পরিস্থিতিকে বলা হয় Race condition।

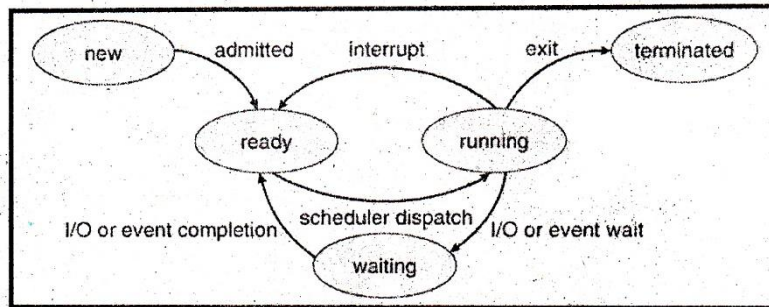
রচনামূলক প্রশ্ন

১। চিত্রসহ Process State বর্ণনা কর।

উত্তরঃ- Program নির্বাহের সাথে সাথে process তার state পরিবর্তন করে থাকে। কোনো Process-এর state দ্বারা তার current বুঝায়। Process execution হলো CPU এবং I/O Bursts-এর একটি Alternating sequence যা CPU burst-এর সাথে শুরু এবং শেষ হয়ে থাকে। সুতরাং, প্রত্যেক Process-ই new, ready, running, blocked or halted-এর যে-কোনো একটি state-এ থাকতে হয়। Ready process হলো এমন একটি Process, যেটি CPU পাওয়ার জন্য waiting করছে। যে Process-টি CPU allocate করে আছে, সেটিকে Running process বলা হয়। কিছু Process আছে, যা আবার Blocked হয়ে থাকে।

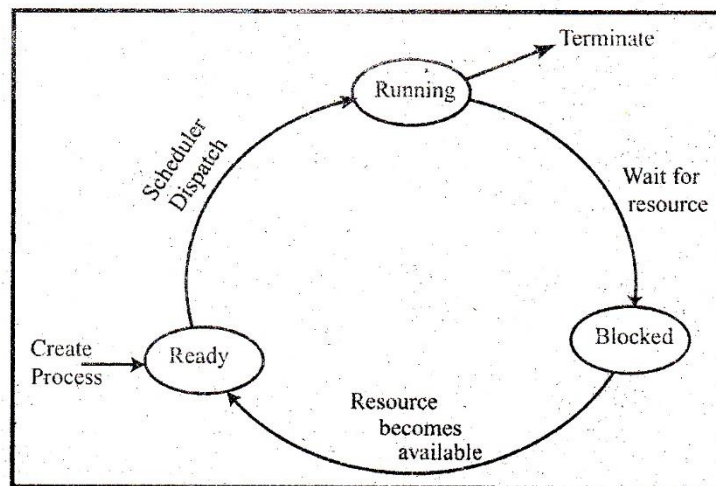
কিছু System-এ Suspended state নামে একটি state যোগ করে থাকে। কোনো process তখনই suspended করা হয়। যখন O/S লক্ষ করে system-এ অনেক process রয়েছে, তখন system-এর load কমানোর জন্য Process suspended করা হয়। কোনো Process-এ সাধারণত নিম্নলিখিত ৫টি State থাকে-

- (i) New, (ii) Running, (iii) Waiting, (iv) Ready, (v) Terminated.



চিত্র: Process state diagram.

Process state diagram (Primitives) নিম্নভাবেও দেখানো যায়ঃ



চিত্র: Process state for primitives processes.

অনুশীলনী-৫

অতিসংক্ষিপ্ত প্রশ্ন

১। সিডিউলিং কাকে বলে?

উত্তরঃ- যখন একাধিক Process রানিং (Running) অবস্থায় যাওয়া জন্য অর্থাৎ নির্বাহের জন্য প্রস্তুত থাকে, তখন Operating system নিজেই সিদ্ধান্ত গ্রহণ করে কোন Process-টি প্রথমে নির্বাহ হবে। Operating system-এর যে অংশ এ সিদ্ধান্ত গ্রহণ করে, তাকে বলা হয় সিডিউলার (Scheduler)।

২। SJF Algorithm-এর বৈশিষ্ট্য লেখ।

উত্তরঃ- SJF Algorithm-এর বৈশিষ্ট্য নিম্নে দেয়া হলো, যথা-

- i) Non-pre-emptive ও pre-emptive পদ্ধতি।
- ii) FCFS Algorithm-এর চেয়ে waiting time কম হয়।
- iii) Optimal পদ্ধতি।
- iv) Possible user abuse.
- v) Must know how long a process will run.

৩। CPU বাউন্ড প্রোগ্রাম ও I/O বাউন্ড প্রোগ্রাম কী?

উত্তরঃ- I/O bound প্রোগ্রামঃ মাল্টিপ্রোগ্রামিং সিস্টেম যে প্রোগ্রামসমূহ প্রসেসিং-এর ক্ষেত্রে বেশিরভাগ সময় I/O অপারেশনের কাজে নিয়োজিত থাকে, সেগুলোকে I/O bound প্রোগ্রাম বলে।

CPU bound প্রোগ্রামঃ- মাল্টিপ্রোগ্রামিং সিস্টেমে যে প্রোগ্রামসমূহ কম সময় I/O অপারেশনে নিয়োজিত থাকে এবং বেশিরভাগ সময় প্রসেসিং-এর কাজে নিয়োজিত থাকে, তাকে CPU bound প্রোগ্রাম বলে।

৪। Turn around time বলতে কী বুঝ?

উত্তরঃ- কোনো জব Submission থেকে শুরু করে Completion (নির্বাহ শেষ) পর্যন্ত যে সময়ের প্রয়োজন, তাকে Turn around time বলে। ভালো Algorithm-এর Turn around time কমানোর প্রয়োজন হয়।

৫। Through put কী?

উত্তরঃ- একক সময়ে নির্বাহ করা (Executed) জবের সংখ্যাই হচ্ছে Through put। প্রতি ইউনিট সময়ে Job নির্বাহের সংখ্যা যত বেশি হবে, সেই Scheduling algorithm তত ভালো হবে।

৬। কয়েকটি Scheduling algorithm এর নাম লেখ।

উত্তরঃ- কয়েকটি Scheduling algorithm-এর নাম নিম্নরূপঃ

- i) FCFS (First Come First Serve) Algorithm.
- ii) SJF (Shortest-Job-First) Scheduling Algorithm.
- iii) Priority Scheduling.
- iv) Round Robin Scheduling.
- v) Multilevel Queue Scheduling.
- vi) Multilevel Feedback Queue Scheduling.

সংক্ষিপ্ত প্রশ্ন

১। CPU-I/O burst cycle বর্ণনা কর।

উত্তরঃ- CPU সিডিউলিং-এর স্বার্থকতা প্রসেসরের বিভিন্ন বৈশিষ্টের উপর নির্ভর করে। I/O Wait cycle এবং execute cycle মিলে প্রসেস নির্বাহ সাইকেল গঠিত হয়। প্রসেস সাধারণত এই দুই স্টেট-এর মধ্যে পরিবর্তিত হয়। CPU Burst-এর মধ্য দিয়ে Process নির্বাহ শুরু হয়। প্রথমে I/O burst এরপর CPU Burst তারপর I/O burst এভাবে ফলো হবে। পরিশেষে শেষ CPU Burst সিস্টেম রিকোয়েস্ট এর মাধ্যমে নির্বাহ বন্ধ হয়।

২। ডিসপ্যাচিং বর্ণনা কর।

উত্তরঃ- Dispatcher হলো এমন একটি মডিউল, যা Short-term-scheduler দ্বারা নির্ধারিত প্রসেসকে CPU-এর নিয়ন্ত্রণ প্রদান করে। Dispatcher নিম্নলিখিত কাজগুলো করে থাকে-

- Switching context
- Switching to user mode
- Jumping to the proper location in the user program.

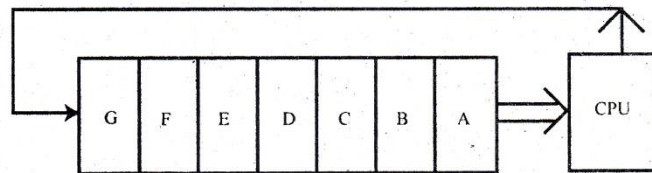
৩। CPU Scheduling সংক্ষেপে বর্ণনা কর।

উত্তরঃ- যখন CPU ফ্রি হয়ে যায়, তখন অপারেটিং সিস্টেম Ready queue থেকে একটি Process select করে। CPU Scheduler প্রসেস নির্ধারণের কাজ করে থাকে। যে সকল Process রেডি কিউতে আছে, সেগুলো থেকে একটিকে নির্দিষ্ট করা হয় এবং CPU-এর নিকট Execution-এর জন্য দেওয়া হয়। উল্লেখ্য যে, Ready queue FIFO নয়। যখন বিভিন্ন সিডিউলিং অ্যালগরিদম ব্যবহার করা হয়, তখন Ready queue FIFO কিউ, Priority Queue অথবা আন অর্ডার লিংক লিস্ট হতে পারে। Ready queue-এর সকল প্রসেস CPU-এর দ্বারা নির্বাহের জন্য অপেক্ষা করতে থাকে। এ সকল প্রসেস থেকে CPU-এর নির্বাহের জন্য প্রসেস নির্ধারণ করাই CPU scheduler-এর কাজ।

রচনামূলক প্রশ্ন

১। Round Robin Scheduling Algorithm বর্ণনা কর।

উত্তরঃ- রাউন্ড রবিন অ্যালগরিদম (Round Robin Scheduling Algorithm): Scheduling Algorithm-এর সবচেয়ে পুরাতন সরল এবং বেশি ব্যবহৃত অ্যালগরিদম হচ্ছে Round Robin (RR)। এ পদ্ধতিতে প্রত্যেক প্রসেসের জন্য নির্দিষ্ট সময় (কোয়ান্টাম) বরাদ্দ থাকে। উক্ত সময়ের মধ্যে কোনো প্রসেসের নির্বাহের কাজ শেষ হলে Process টি সরানো হয় এবং পরবর্তী প্রস্তুত প্রসেস একইভাবে উক্ত সময় পর্যন্ত নির্বাহের সুযোগ পায়। যদি Blocked কোনো Process নির্বাহে ত্রুটি দেখা দেয় কিংবা প্রসেসের নির্দিষ্ট কোয়ান্টাম অতিবাহিত হয়ে যায়, তাহলে প্রসেসটি Ready queue তালিকা সর্বশেষে স্থান পায় এবং তার পূর্বের Process-গুলোর নির্বাহের কাজ শেষ হলে পুনরায় উক্ত প্রসেসটি নির্বাহের সুযোগ পায়। যদি কোনো Process Blocked অবস্থা প্রাপ্ত হয় অথবা নির্দিষ্ট সময়ের (কোয়ান্টামের) পূর্বেই Process-এর নির্বাহের কাজ শেষ হয়, তাহলে সেক্ষেত্রে CPU-এর Switching কাজ সম্পন্ন করতে হয়।



চিত্র : Round Robin পদ্ধতি

অনুশীলনী-৬

অতিসংক্ষিপ্ত প্রশ্ন

১। **Deadlock** কী?

উত্তরঃ- মাল্টিপ্রোগ্রামিং (Multiprogramming) সিস্টেমে কয়েকটি Process নির্দিষ্ট কিছু সংরক্ষিত Resource ব্যবহারের জন্য চেষ্টা করে। এক্ষেত্রে যদি ঐ সময় কোনো Process-এর Request-এ Resource-গুলো প্রাপ্যতাবিহীন নয়, তবে Process টি অপেক্ষমাণ অবস্থায় (Waiting state) প্রবেশ করে। এরূপ ক্ষেত্রে অনেক সময় অপেক্ষমাণ Process টি আর কখনো তার অবস্থার পরিবর্তন করতে পারে না। কারণ যে Resource টি ব্যবহারের জন্য Request করেছে, সে Resource টি অপেক্ষমাণ Process-গুলো দ্বারা ধরা থাকে। এ অবস্থাকে Deadlock বলে।

২। **Preemptive resource** কী?

উত্তরঃ- যদি কোনো Resource-এর ক্ষেত্রে এমন হয় যে, Process execution থাকা অবস্থায় Priority-এর ভিত্তিতে অন্য Process-এর execution সম্পন্ন করতে হয়, তবে এ ধরনের Resource-কে Preemptive resource বলে। মেমরি (Memory) হলো Preemptive resource-এর উদাহরণ।

৩। **Non-preemptive resource** কী?

উত্তরঃ- Process চলাকালীন যদি কোনো Resource-এর জন্য Request পাঠানো হয় এবং ঐ Resource-এ বিদ্যমান Process নির্বাহ সম্পন্ন না হওয়া পর্যন্ত Resource টি মুক্তি না পায়, তবে সে সকল Resource-কে Non-preemptive resource বলে।

৪। **Deadlock-এর শর্তাবলী** কী কী?

উত্তরঃ- (i) Mutual exclusion.

(ii) Hold & wait.

(iii) No-preemption.

(iv) Circular wait.

৫। **Deadlock detection algorithm** কত প্রকার ও কী কী?

উত্তরঃ- সাধারণত Deadlock detection algorithm দু'ধরনের হয়। যথা-

(i) Single instance of each resource type এবং

(ii) Several instances of a resource type.

৬। **Deadlock recovery-তে** কয়টি **Option** আছে ও কী কী?

উত্তরঃ- এক্ষেত্রে Deadlock breaking-এ দু'ধরনের Options রয়েছে। যথা-

(i) Circular wait-কে ভেঙ্গে দেয়ার জন্য এক/একাধিক Process-কে kill বা abort করা।

(ii) Deadlock process থেকে এক/একাধিক Resource preempt করা।

৭। **Roll back** বলতে কী বুঝ?

উত্তরঃ- কোনো রিসোর্স বা প্রসেস Preempt করার পর উক্ত রিসোর্স/প্রসেসকে Normal execution-এর উদ্দেশ্যে Safe state-এ নিয়ে যাওয়া এবং একে ঐ State থেকে Restart করে এর Resuability নিশ্চিত করা।

৮। কী কী কারণে Deadlock হয়?

উত্তরঃ- চারটি শর্তে একটি সিস্টেমে ডেডলক ঘটতে পারে।

- (i) পারস্পরিক বর্জন (Mutual exclusion)।
- (ii) Hold and wait
- (iii) No preemption
- (iv) Circular wait.

সংক্ষিপ্ত প্রশ্ন

১। একটি System Deadlock-এ পড়বার শর্তগুলো বর্ণনা কর।

উত্তরঃ- যুগপৎভাবে সংঘটিত চারটি শর্তে একটি সিস্টেমে ডেডলক অবস্থা ঘটতে পারে। শর্তগুলো হলো-

- (i) পারস্পরিক বর্জন (Mutual exclusion)ঃ এ শর্তমতে কমপক্ষে একটি রিসোর্স নন-শেয়ারাবল মোডে থাকবে অর্থাৎ, কেবলমাত্র একটি প্রসেস একই সঙ্গে একটি রিসোর্স ব্যবহার করতে পারবে। যদি অন্য প্রসেস ঐ একই রিসোর্স ব্যবহারের অনুরোধ করে, তবে অনুরোধকৃত পরের প্রসেসটিকে রিসোর্স মুক্ত না হওয়া পর্যন্ত অপেক্ষা করতে হবে। তা না হলে ডেডলক সংঘটিত হবে।
- (ii) Hold and wait : কমপক্ষে একটি রিসোর্স একটি প্রসেস-এর দখলে থাকবে এবং অতিরিক্ত রিসোর্সের অপেক্ষায় থাকবে, যা বর্তমানে অন্য প্রসেস-এর দখলে আছে। তবে ডেডলক অবস্থার সৃষ্টি হবে।
- (iii) No preemption : সিস্টেমের Resource-গুলোকে অগ্রাধিকার বলে দখল দিতে পারবে না। অর্থাৎ, কোনো প্রসেস কর্তৃক দখলকৃত কোনো রিসোর্স শুধুমাত্র প্রসেসটির কাজ সম্পন্ন হওয়ার পর স্বতঃপ্রস্তুত হয়ে মুক্ত করতে পারবে। অন্যথায় ডেডলক অবস্থা ঘটবে।
- (iv) Circular wait : এক সেট ($P_0, P_1, P_2, \dots, P_n$) অপেক্ষমাণ প্রসেস এমনভাবে থাকবে যে, P_0 অপেক্ষা করবে P_1 কর্তৃক দখলকৃত রিসোর্সের জন্য, P_1 অপেক্ষা করবে P_2 কর্তৃক দখলকৃত রিসোর্সের জন্য, P_2 অপেক্ষা করবে P_3 কর্তৃক দখলকৃত রিসোর্সের জন্য, P_{n-1} অপেক্ষা করবে P_n কর্তৃক দখলকৃত রিসোর্সের জন্য এবং P_n অপেক্ষা করবে P_0 কর্তৃক দখলকৃত রিসোর্সের জন্য। অর্থাৎ, বৃত্তীয় ক্রমানুসারে প্রসেসগুলো রিসোর্স ব্যবহারের অপেক্ষায় থাকে, তবে ডেডলক ঘটবে।

২। Deadlock থেকে মুক্ত হওয়ার উপায়সমূহ কী কী?

উত্তরঃ- সাধারণত তিনটি পদ্ধতিতে ডেডলক সমস্যা মোকাবিলা করা হয়-

- (i) ডেডলক প্রতিরোধ করার জন্য বা ডেডলক এড়ানোর জন্য আমরা এমন প্রটোকল ব্যবহার করতে পারি যেন সিস্টেমে কোনো ডেডলক অবস্থা না আসে।
- (ii) সিস্টেম ডেডলক অবস্থা গ্রহণ করে, চিহ্নিত করে এবং তা পুনরুদ্ধার করে।
- (iii) আমরা সর্বাংশে ডেডলক সমস্যা অবহেলা করতে পারি এবং এমন ভান করতে পারি যেন সিস্টেমে কোনো ডেডলক অবস্থা ঘটবেই না।

৩। Deadlock prevention বলতে কী বুঝ?

উত্তরঃ- ডেডলক প্রতিরোধকরণ ব্যবস্থায় একসেট মেথড এমনভাবে যোগান দেয়া হয় যেন ডেডলক শর্তাবলির (Deadlock condition) কোনো একটি শর্তও না ঘটে। মেথডগুলো এমনভাবে ডিজাইন করা হয় যেন রিসোর্স ব্যবহারে প্রসেস কর্তৃক অনুরোধ সঠিক সময়ে হয়।

রচনামূলক প্রশ্ন

১। **Deadlock** দূরীকরণের **Algorithm** বর্ণনা কর।

উত্তরঃ- Deadlock avoidance algorithm গুলোর মধ্যে Safe state algorithm, Resource allocation graph algorithm, Banker's algorithm উল্লেখযোগ্য।

নিম্নে Safe state algorithm ও Resource graph algorithm বর্ণনা করা হলো-

Safe state algorithm: Deadlock সমস্যা Avoid করার জন্য Scheduling algorithm হলো Dijkstra's banker's algorithm. (It is modeled after the lending policies often employed in banks. This algorithm could be used in a banking system to ensure that the bank never allocates its available cash in such a way that it can no longer satisfy the needs at all its customers).

যখন কোনো নতুন Process, System-এ প্রবেশ (Enter) করে, তখন সে নতুন Process-কে অবশ্যই তার প্রত্যেক Resource-এর জন্য প্রয়োজনীয় (Needed) সংখ্যক Instance declare করতে হবে, কিন্তু তা অবশ্যই System-এর মোট Resource-এর চেয়ে বেশি হতে পারবে না। যখন কোনো User কিছু Resource-এর জন্য Request করে, তখন অবশ্যই দেখতে হবে Request resource-গুলোর মাধ্যমে System-টি Safe state-এ যাবে কি না। যদি তাই হয়, তাহলে Resource-গুলো Allocated করা হবে, অন্যথায় Process-কে অবশ্যই অপেক্ষা করতে হবে যতক্ষণ না পর্যন্ত Allocated-গুলোকে অন্য Process মুক্ত (free) না করে। সাধারণত যখন কোনো System প্রতিটি Process-এর জন্য Resource বরাদ্দ করতে পারবে, তখনই একে Safe state বলে ধরা হবে। যতক্ষণ পর্যন্ত System টি Safe state-এ থাকবে, ততক্ষণ Deadlock avoidance করা যাবে, কিন্তু State unsafe হলেই Deadlock সৃষ্টি হবে এমন কোনো কথা নেই।

ধরা যাক, একটি System-এ 12টি Tape drive এবং 3টি Process বিদ্যমান। Process P_0 এর জন্য 10টি, P_1 -এর জন্য 4টি এবং P_2 -এর জন্য 9টি Tape drive প্রয়োজন হতে পারে। T_0 সময়ে P_0 5টি, P_1 2টি, ও P_2 2টি করে Tape drive hold আছে। এমতাবস্থায় 3টি Tape drive free আছে এবং প্রত্যেকটি Process-ই তাদের Request-কে Satisfy করছে। এরূপ -কে বলে। কিন্তু প্রসেস-এর জন্য প্রয়োজনীয় Resource-এর সংকট ঘটলেই Unsafe state condition সৃষ্টি হবে।

Banker's algorithm বাস্তবায়নের জন্য নিম্নলিখিত Data structure অবশ্যই Maintain করতে হবে-

Data structures:

Let n = number of processes, and m = number of resource types.

Available: Vector of length m . If $available[j] = k$, there are k instances of resource type R_j Available.

Max: $n \times m$ matrix. If $allocation[i, j] = k$, then P_i may request at most k instances of resource type R_j .

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. $Need[i, j] = Max[i, j] - Allocation[i, j]$

নিম্নে Safety algorithm দেয়া হলো-

Safety algorithm:

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

Initialize.

Work: = Available and Finish $[i] = \text{false}$.

for $i = 0, 1, 2, 3, \dots, n$

(ii) Find an i such that both

a. $\text{Finish}[i] = \text{false}$

b. $\text{Need}[i] \leq \text{work}$

If no such i exists go step 4.

(iii) $\text{Work} = \text{Work} + \text{Allocation}[i]$.

$\text{Finish}[i] = \text{true}$.

Go to step 2.

(iv) If $\text{Finish}[i] = \text{true}$ for all i , then the system is in a safe state. This algorithm may require an order of $m \times n^2$ operation to determine whether a state is safe.

Resource-request algorithm (Banker's algorithm):

ধরি, P_i = Number of processes

R_j = Number of resources

k = Requests demanded by process.

যদি Request i (j) = k হয়, তখন Process P_i , Resource $R[j]$ -এর জন্য k সংখ্যক Request পাঠাবে।

Since process has executed its maximum claims.

(i) If Request $i \leq \text{need}[i]$, go to step 2, অন্যথায় Error সংঘটিত হবে।

(ii) If Request $i \leq \text{Available}$, go to step 2. অন্যথায় Process P_i waiting থাকবে। Since resource are not available.

(iii) $\text{Available} = \text{Available} - \text{Request}[i]$;

$\text{Allocation}[i] = \text{Allocation}[i] + \text{Request}[i]$;

$\text{Need}[i] = \text{Need}[i] - \text{Request}[i]$;

যদি Result টি Safe state-এর হয়, তবে Process P_i resource-সমূহে Allocate হবে, অন্যথায় P_i অপেক্ষা করবে।

অনুশীলনী-৭

অতিসংক্ষিপ্ত প্রশ্ন

১। ফ্রাগমেন্টেশন কী?

উত্তরঃ- পার্টিশন অ্যালোকেশনের সবচেয়ে বড় সমস্যাটির নাম ফ্রাগমেন্টেশন। যখন কোনো পার্টিশনে প্রোগ্রামের কাজ শেষ হয়ে যায়, তখন নতুন কোনো প্রোগ্রামকে উক্ত স্থানে লোড করার ক্ষেত্রে যদি প্রোগ্রামটির দৈর্ঘ্য বেশি হয়, তাহলে তাকে স্মৃতিতে লোড করা যায় না। আবার প্রোগ্রামিং দৈর্ঘ্য উক্ত পার্টিশনের তুলনায় কম হলে অতি সামান্য অংশ ব্যবহৃত হয় এবং বাকি অংশে অন্য কোনো প্রোগ্রাম লোড করা যায় না। এ অব্যবহৃত অংশই ফ্রাগমেন্ট এবং স্মৃতিতে এ অব্যবহৃত ফ্রাগমেন্টের সংখ্যা বৃদ্ধি পাওয়াকে ফ্রাগমেন্টেশন বলা হয়।

২। Fragmentation কত প্রকার ও কী কী?

উত্তরঃ- Fragmentation ২ প্রকার, যেমন-

- (i) Internal fragmentation
- (ii) External fragmentation.

৩। পেজিং কী?

উত্তরঃ- স্মৃতি অ্যালোকেশনের একটি পদ্ধতি যা দ্বারা প্রোগ্রামকে নির্দিষ্ট দৈর্ঘ্যে বিভক্ত করা হয়। প্রোগ্রামের এই অংশসমূহকে পেজ বলা হয় এবং একইভাবে কোর স্মৃতিকে সমান দৈর্ঘ্যে বিভক্ত করা হয়, যা ব্লক নাম পরিচিত। পেজসমূহ প্রসেসিং-এর সময় ব্লকে লোড হয়।

৪। Swapping কী?

উত্তরঃ- Program-কে Main memory ও Secondary memory-এর মধ্যে স্থানান্তর করার প্রক্রিয়াকে Swapping (সোয়াপিং) বলে।

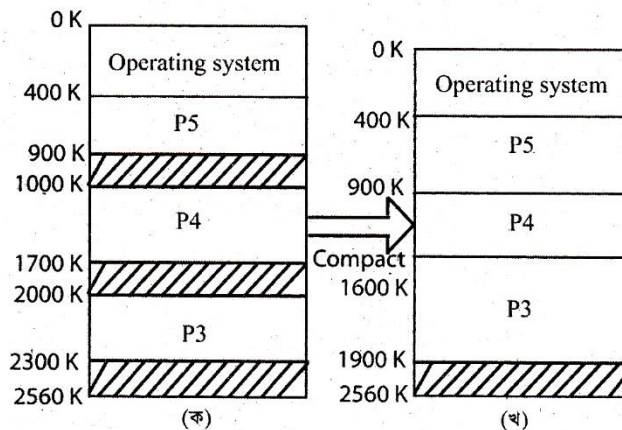
৫। Memory Management বলতে কী বুঝায়?

উত্তরঃ- যে সকল উপায় বা কৌশল অবলম্বন করে মেমরির সঠিক ব্যবহার করা যায়, তাকে Memory Management কৌশল বলে।

সংক্ষিপ্ত প্রশ্ন

১। Internal ও External fragmentation-এর বর্ণনা দাও।

উত্তরঃ- External fragmentation সমস্যা দূরীভূত করার একটি পদ্ধতি হচ্ছে “Compaction”। এ পদ্ধতিতে সমস্ত Free memory space-সমূহকে একত্রিত করে একটি Large block-এ রূপান্তর বলা হয়।



চিত্র (ক) লক্ষ করলে দেখা যায় যে তিনটি Free memory space (Hole) 100K, 300K এবং 260K বিদ্যমান, যা পরবর্তী চিত্রে Compact করে একটি Hole-এ দেখানো হয়েছে। তবে Compaction সবসময় সম্ভব নয়। কারণ যদি Process এবং Internal address-গুলো Dynamically এবং Execution time-এ Relocate করা যায়, তবেই এটি সম্ভব। আর যদি Process এবং Internal address-গুলোর Relocation statically এবং Load time-এ সংঘটিত হয়, তবে Compaction সম্ভব নয়।

২। ভার্চুয়াল মেমরি বলতে কী বুঝ?

উত্তরঃ- মাল্টিপ্রোগ্রামিং ও মাল্টিপ্রসেসিং সিস্টেমে একাধিক জব একসঙ্গে প্রসেস করতে হয় বলে একাধিক জব বা প্রোগ্রাম মেইন মেমরিতে থাকতে হয়। এজন্য মেইন মেমরি খুব বড় হওয়া প্রয়োজন। মেইন মেমরির স্বল্পতা বা সীমাবদ্ধতা দূরীকরণের জন্য ভার্চুয়াল স্টোরেজ সিস্টেম ব্যবহৃত হয়। ভার্চুয়াল মেমরি হলো মেইন মেমরি ও কোনো প্রত্যক্ষ সঞ্চয় মাধ্যমের এমন সমাহার, যাকে ভার্চুয়াল স্টোরেজ অপারেটিং সিস্টেম দ্বারা পরিচালনা করলে তা ব্যবহারকারীর নিকট প্রায় অসীম মেইন মেমরির মতো মনে হয়।

৩। সেগমেন্টেশন কী?

উত্তরঃ- Segment হচ্ছে তথ্যের সমষ্টি, যা একটি Entity'র ন্যায় আচরণ করে। এটি এক বা একাধিক Program, একটি Database অথবা একাধিক Database-ও হতে পারে। Segmented পরিবেশে Address-সমূহ দ্বি-মাত্রিক (Two dimensional) হয় এবং Address দুটি Component দ্বারা প্রকাশ করা হয়-

(i) সেগমেন্টের নাম (Name of segment)

(ii) সেগমেন্টের অফসেট (Offset of segment)

এক্ষেত্রে প্রতিটি Process-এর নিজস্ব Segment table থাকে। যখন কোনো Process-এর সকল Segment, Main memory-তে, Load হয়, তখন প্রসেসের জন্য Segment table তৈরি হয় এবং Main memory-তে Load হয়। প্রত্যেক Segment table-এর Entity প্রধান মেমরিতে কোনো নির্দিষ্ট Segment-এর Starting address এবং Segment-এর দৈর্ঘ্য ধারণ করে। কোনো Word-কে Memory থেকে পড়ার ক্ষেত্রে Virtual অথবা Logical address অনুবাদ করে (Segment নং এবং offset) Segment table-এর সাহায্যে Physical address তৈরি করে। যেহেতু Segment table variable length-এর হয় (প্রসেসের দৈর্ঘ্যের উপর নির্ভরশীল), সেজন্য একে Address করার জন্য Main memory-তে নিতে হয়।

৪। Demand paged allocation-এর সুবিধা ও অসুবিধা কী?

উত্তরঃ- নিম্নে ডিমান্ড পেজড অ্যালোকেশন পদ্ধতির সুবিধাগুলো প্রদত্ত হলোঃ

(i) স্বল্প দৈর্ঘ্যের কোর স্মৃতিকে অধিক দৈর্ঘ্যের প্রোগ্রাম নির্বাহ করা যায়।

(ii) অপারেটরের মধ্যস্থতার প্রয়োজন হয় না।

(iii) সিস্টেমের Throughput বৃদ্ধি করে।

(iv) প্রোগ্রাম কোর স্মৃতিকে বৃহৎ স্মৃতি হিসেবে কল্পনা করতে পারে এবং নিজের প্রোগ্রামকে মাল্টিপ্রোগ্রামিং পরিবেশে নির্বাহ করতে পারে।

(v) টাইম শেয়ারিং পদ্ধতির জন্য এটি বেশি উপযোগী।

অসুবিধা (Disadvantage):

যখন অধিক সংখ্যক প্রোগ্রাম টাইম শেয়ার করে স্বল্প দৈর্ঘ্যের কোর স্মৃতিকে ব্যবহার করে অথবা পেজিং জাজমেন্ট অ্যালগরিদম অপেক্ষাকৃত নিম্নমানের পেজকে কোর স্মৃতিতে নিয়ে পড়া, স্থানান্তর করা এবং পুনরায় সহায়ক স্টোরেজ লোড করার ক্ষেত্রে সিস্টেমের Throughput অত্যন্ত কমে যায়।

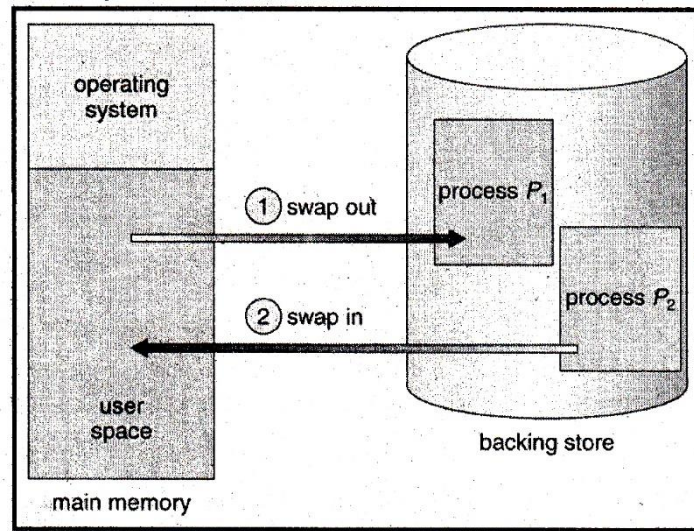
রচনামূলক প্রশ্ন

১। চিত্রসহ **Swapping** প্রসেস বর্ণনা কর।

উত্তরঃ- বিভিন্ন System-এর Memory management বিভিন্ন রকম হয়ে থাকে। Multiprogramming ব্যবস্থায় কিছু Program প্রধান মেমরিতে (Main Memory) অবস্থান করে এবং অন্যান্য Program-গুলো সহায়ক মেমরিতে (Secondary memory/Disk storage/Backing storage) অবস্থান করে। Main Memory থেকে নিয়ে Program-কে যখন কোনো CPU নির্দিষ্ট সময় পর্যন্ত process করে, তখন অন্য Program প্রস্তুত অবস্থায় (Ready condition) আসে এবং প্রসেসকৃত অব্যবহৃত Program-কে Main Memory থেকে Secondary Memory-তে নিয়ে যাওয়া হয়। এভাবে Program-কে Main Memory ও Secondary memory (Backing store)-এর মধ্যে স্থানান্তর করার প্রক্রিয়াকে সোয়াপিং (Swapping) বলা হয়।

সোয়াপিং প্রক্রিয়া (Swapping process) :

নিম্নে চিত্রের সাহায্যে Swapping প্রক্রিয়া ব্যাখ্যা করা হলো-



অনেক ক্ষেত্রেই অগ্রগণ্যতার ভিত্তিতে Swapping প্রক্রিয়া ঘটে। যদি কখনো Memory manager, Lower priority process নিয়ে কাজ করে এমন অবস্থায় কোনো Higher priority process যদি উপস্থিত হয়, তখন Memory manager, Lower priority process-কে Swap করে এবং Higher priority process-কে Execute করে। Execute শেষ হলেই সে আবার Lower priority process execute করতে থাকে। এ ধরনের Swapping process-কে Roll out এবং Roll in বলা হয়।

Swapping-এর জন্য Backing store অথবা First disk storage ব্যবহার করা হয়। এ Disk storage টি Large memory বিশিষ্ট হতে হয়। কারণ, এখানেই সকল User-এর সকল Process সংরক্ষিত থাকে। Process-গুলো Backing memory dispatcher-এ একটি Ready queue-তে অবস্থান করে। যখন Main memory'র Process execution শেষ হবে, তখন Main memory dispatcher-কে পরবর্তী Process-এর জন্য Call করে। Dispatcher প্রধান memory থেকে নির্বাহকৃত Process-কে swap out করে এবং পরবর্তী Process, প্রধান memory-তে swap in হয়। এভাবেই swap in এবং swap out প্রক্রিয়ার মাধ্যমে Swapping প্রসেস চলতে থাকে।

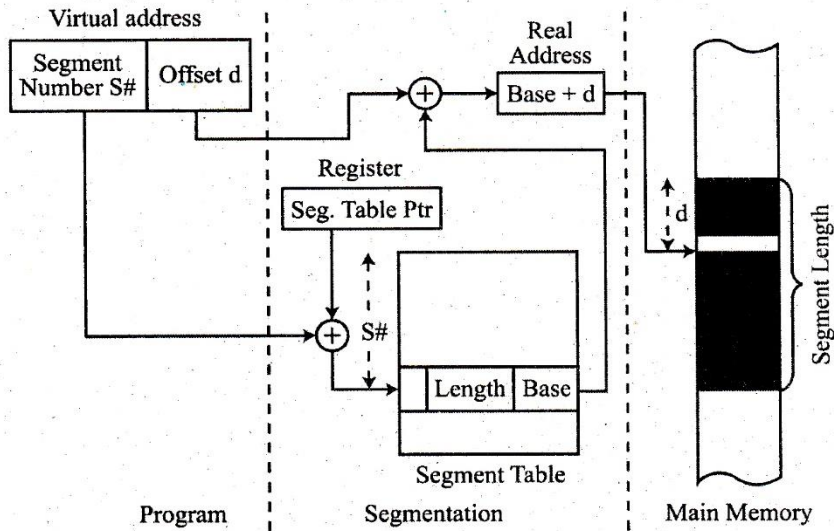
২। চিত্রসহ **Segmented allocation** বর্ণনা কর।

উত্তরঃ- Segment হচ্ছে তথ্যের সমষ্টি, যা একটি Entity'র ন্যায় আচরণ করে। এটি এক বা একাধিক Program, একটি Database অথবা একাধিক Database-ও হতে পারে। Segmented পরিবেশে Address-সমূহ দ্বি-মাত্রিক (Two dimensional) হয় এবং Address দুটি Component দ্বারা প্রকাশ করা হয়-

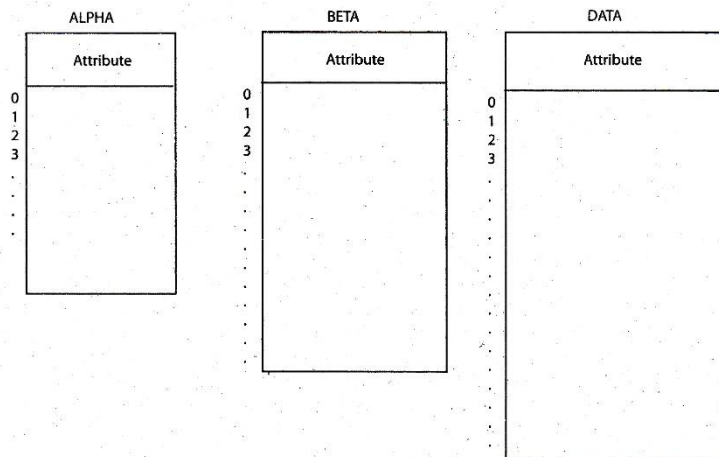
(i) সেগমেন্টের নাম (Name of segment)

(ii) সেগমেন্টের অফসেট (Offset of segment)

এক্ষেত্রে প্রতিটি Process-এর নিজস্ব Segment table থাকে। যখন কোনো Process-এর সকল Segment, Main memory-তে, Load হয়, তখন প্রসেসের জন্য Segment table তৈরি হয় এবং Main memory-তে Load হয়। প্রত্যেক Segment table-এর Entity প্রধান মেমরিতে কোনো নির্দিষ্ট Segment-এর Starting address এবং Segment-এর দৈর্ঘ্য ধারণ করে। কোনো Word-কে Memory থেকে পড়ার ক্ষেত্রে Virtual অথবা Logical address অনুবাদ করে (Segment নং এবং offset) Segment table-এর সাহায্যে Physical address তৈরি করে। যেহেতু Segment table variable length-এর হয় (প্রসেসের দৈর্ঘ্যের উপর নির্ভরশীল), সেজন্য একে Address করার জন্য Main memory-তে নিতে হয়।



প্রত্যেক Segment-এরই নিজস্ব Access অধিকার সংশ্লিষ্ট থাকে। Segmentation, Dynamic linking সুযোগ-সুবিধা সম্পাদন করে থাকে। অর্থাৎ, যখন একটি Caller program কর্তৃক অন্য একটি Call statement নির্বাহের প্রয়োজন হয়, তখন Loading mechanism ঐ Call statement-কে খুঁজে বের করে Core memory-তে Load করে এবং Calling program বা Call statement-কে Caller-এর সাথে একত্রে বন্ধন সৃষ্টি করে। একে Dynamic linking বলা হয়।



উপরের চিত্রে Programmer-এর দৃষ্টিতে Segmented address space দেখানো হলো। প্রত্যেকটি Segment-ই একটি স্বতন্ত্র (Distinct) ও আলাদা লিনিয়ার মেমরি এবং প্রত্যেকটি Segment-এর আলাদা আকার (Size) আছে। অর্থাৎ, প্রত্যেকটি Segment-এর Size এক রকম হওয়ার প্রয়োজন নেই।

ধরা যাক, ALPHA ও BETA segment-দ্বয় নির্বাহ ও DATA Segment শুধুমাত্র ডাটা ধারণ করেছে। চিত্র হতে আর দেখা যাচ্ছে যে, Segment-গুলোর আকার (Size) ভিন্ন রকম।

আমরা জানি যে, দ্বি-মাত্রিক Segmented system-এ প্রত্যেক Segment-এর দু'টো অংশ থাকে- একটি হলো Segment-এর নাম এবং অপরটি হলো ঐ Segment-এর মধ্যস্থিত Offset.

যেমন-

LOAD <DATA> |25

এর অর্থ হলো DATA নামের Segment-এর মধ্যস্থিত Location 25 হতে একটি WORD LOAD করা।

আবার, TRANSFER <BETA> |7

এর অর্থ হলো BETA নামের Segment-এর মধ্যস্থিত Location 7-একটি WORD স্থানান্তর (Transfer) করা।

উপরের Instruction দুটিতে “|” প্রতীকের দ্বারা একটি Address-এর Segment নাম ও Offset আলাদা করা হয়েছে। বাস্তবে Segment অ্যাড্রেস স্পেস সিস্টেমে সেগমেন্টসমূহ নামের পরিবর্তে সংখ্যা (Number) দ্বারা প্রকাশ করা হয়। ঠিক যেমন প্রচলিত অ্যাসেম্বলি ল্যাংগুয়েজে লোকেশনকে প্রতীক দ্বারা প্রকাশ করা হয়। যেমন-

TRANSFER <BETA> | LOOP + 2 স্ট্রাকশনটি যেভাবে নির্বাহ (Executed) হবে।

0500 3126 এখানে $26 = 24 + 2$; 24 হলো LOOP-এর মান, BETA হলো সেগমেন্ট নম্বর, যা 3 দ্বারা প্রকাশ করা হয়েছে, LOOP হলো BETA-এর মধ্যে অবস্থিত 24 লোকেশন। 0500 হলো TRANSFER ইনস্ট্রাকশনের মেশিন ল্যাংগুয়েজ কোড।

অনুশীলনী-৮

অতিসংক্ষিপ্ত প্রশ্ন

১। SASD এবং DASD-এর মধ্যে পার্থক্য কী?

উত্তরঃ- SASD ও DASD-এর মধ্যকার পার্থক্য নিম্নে দেওয়া হলো-

ক) SASD (Serial Access Storage Device)-এ Information জমা করার জন্য অনুবর্তিতা বা অনুক্রমিকা অনুযায়ী অবস্থান বা প্রবেশ করতে হয়, কিন্তু DASD (Direct access Storage Device)-এ অনুক্রমিতার প্রয়োজন হয় না।

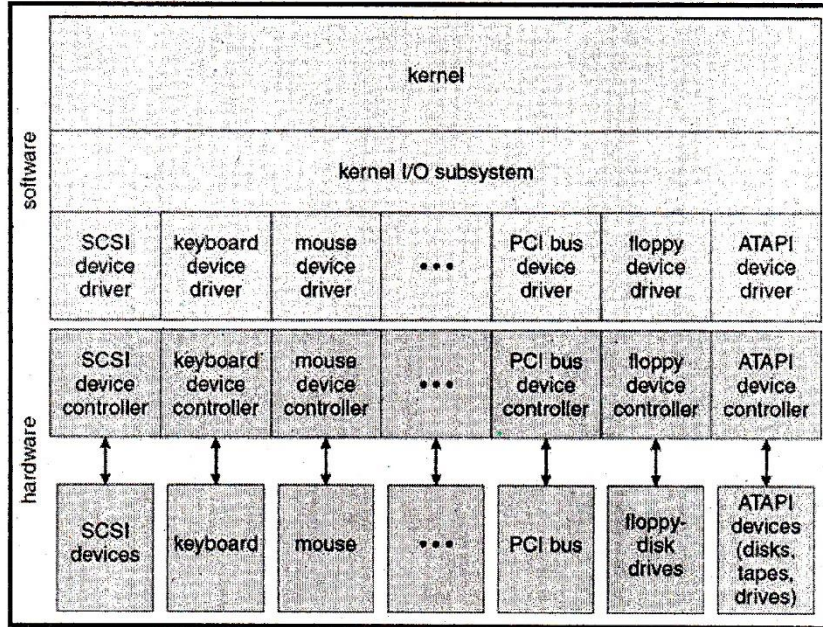
খ) SASD-এ প্রথম হতে দূরের কোনো Record-এ যাওয়ার জন্য মধ্যবর্তী কিছু Record skip করতে হয়, কিন্তু DASD-এর Record skip করার প্রয়োজন নেই।

গ) SASD-এর চেয়ে DASD -এ ডাটা Read/write করতে কম সময় লাগে।

সংক্ষিপ্ত প্রশ্ন

১। I/O Software-এর Goal বর্ণনা কর।

উত্তরঃ- মূলত I/O Hardware বা Device-সমূহ ক্যাবল কিংবা বায়ু মাধ্যমে Signal পাঠিয়ে Computer-এর সাথে যোগাযোগ প্রতিষ্ঠা করে। Device-টি machine-এর সাথে একটি সংযোগস্থল (Connection point), যাকে Port-ও বলে তার মাধ্যমে যোগাযোগ করে। তবে Bus মাধ্যমেও Device-সমূহ Computer-এর সাথে যোগাযোগ করতে পারে।



I/O System বলতে I/O Hardware ও I/O Software উভয়কেই বুঝায়। অর্থাৎ, I/O Hardware ও I/O Software নিয়েই I/O System গঠিত হয়।

সাধারণত Physical যন্ত্রপাতিগুলো হলো I/O Hardware অর্থাৎ, I/O Hardware হচ্ছে Computer-এর Input-output অপারেশন কম্পিউটারের সাথে যুক্ত Device-সমূহ। I/O Software হচ্ছে এ সকল I/O Device-সমূহের ব্যবস্থাপনা ও নিয়ন্ত্রণের জন্য প্রয়োজনীয় Program বা set of program (প্রোগ্রাম সমষ্টি)। I/O Hardware-সমূহের প্রাণ সঞ্চার করতে I/O Software-সমূহের বিকল্প নেই। I/O Software ছাড়া I/O Hardware-সমূহ অচল। (I/O Software-সমূহ I/O Hardware-এর জন্য সার্বিক service প্রদান করে, যেমন-

- (i) আই/ও সিডিউলিং (I/O Scheduling)
- (ii) বাফারিং (Buffering)

(iii) স্পুলিং (Spooling)

(iv) ভুল মোকাবেলা (Error handling)

(v) ডিভাইস রিজার্ভেশন (Device Reservation)

এছাড়াও Input/output hardware-সমূহের নিয়তই পরিবর্তন ঘটছে, নতুন নতুন criteria নিয়ে বাজারে আসছে Input/output hardware-এ সকল Hardware-সমূহের Criteria নির্ধারণসহ কম্পিউটারের যাবতীয় কাজের সাথে Input/output hardware-সমূহের কাজের সমন্বয় বিধান করা, Hardware-সমূহের যাবতীয় বিষয়াদি পর্যবেক্ষণ, নিয়ন্ত্রণ ও ব্যবস্থাপনা করাই Input/output software-এর মূল লক্ষ্য ও উদ্দেশ্য।

অনুশীলনী-৯

অতিসংক্ষিপ্ত প্রশ্ন

১। ফাইল কী?

উত্তরঃ- পরস্পর সম্পর্কযুক্ত কিছু সংখ্যক তথ্য বা রেকর্ডকে যখন একটি ইউনিট হিসেবে বিবেচনা করা হয় অথবা পরস্পর সম্পর্কযুক্ত কিছু সংখ্যক ডাটা ইউনিটের সমবেশ, যেগুলো বিভিন্ন Secondary storage device, যেমন- magnetic disk, magnetic tape, optical disk ইত্যাদি সংরক্ষিত থাকে, তাদেরকে ফাইল (file) বলে। Computer-এ বিভিন্ন ধরনের file রয়েছে, যেমন- txt file, doc file etc.

২। ফাইল সিস্টেম বলতে কী বুঝায়?

উত্তরঃ- তথ্যের লজিক্যাল অর্গানাইজেশনের সাথে ফাইল সিস্টেম সম্পর্কযুক্ত। একটি কম্পিউটার সিস্টেমের ফিজিক্যাল ডাটা স্টোরেজ ডিভাইসে ডাটা স্টোরেজ পুনরুদ্ধার করা সংক্রান্ত কাজসমূহ অপারেটিং সিস্টেমের যে অংশ সম্পাদনা করে, তাকে ফাইল সিস্টেম বলা হয়।

৩। ফাইল পয়েন্টার কী?

উত্তরঃ- ফাইল read বা write operation-এ ব্যবহৃত pointer টিই ফাইল পয়েন্টার। সাধারণ অবস্থায় read বা write operation-এর সময় pointer ফাইলের শুরুতে থাকে। ইচ্ছে করলে এ pointer টি ব্যবহার করে file-এর যে কোনো location থেকে আমরা read বা write operation সম্পন্ন করতে পারি।

৪। Extension সহ বিভিন্ন ধরনের File-এর নাম লেখ।

উত্তরঃ- Executable (.exe) file, Object file (.obj), source file (.c, .p, .pas, .asm), Text file (.txt, .doc) ইত্যাদি।

৫। ফাইল অপারেশনগুলো লেখ।

উত্তরঃ- File নিয়ে বিভিন্ন ধরনের কাজ করা যায়। নিম্নলিখিত অপারেশনগুলো উল্লেখ করা হলো-

(i) ফাইল তৈরিকরণ (Creating a file)

(ii) ফাইলে লিখন (Writing a file)

(iii) ফাইল পঠন (Reading a file)

(iv) ফাইল খুঁজে বের করা (Searching a file)

(v) ফাইল মুছে ফেলা (Deleting a file)

(vi) ফাইল ছাঁটাই করা (Truncating a file)।

৬। Bit vector বলতে কী বুঝায়?

উত্তরঃ- ডিস্কের Free space management-এর একটি পদ্ধতির নাম হলো Bit vector। এ পদ্ধতিতে disk-এর Free space list-কে Bit map বা Bit vector দ্বারা Implement করা হয়। যদি disk-এর কোনো Block free হয়, তবে তাকে 1; আর যদি Block টি allocated হয়, তবে তাকে 0 (Zero) দ্বারা নির্দিষ্ট করা হয়।

সংক্ষিপ্ত প্রশ্ন

১। Database Management System ও File Processing System-এর মধ্যে পার্থক্য লিখ।

উত্তরঃ- ডাটাবেস ম্যানেজমেন্ট সিস্টেমঃ কতকগুলো সম্বন্ধযুক্ত (Interrelated) ডাটার একত্রীকরণ ও ঐ ডাটাকে Access করার জন্য যে program ব্যবহৃত হয়, এসব কাজকে একত্রে Database Management System বলা হয়। কতকগুলো সম্পর্কযুক্ত ডাটার একত্রীকরণকে Database বলা হয়। (Organized collection of data stored in a file that is database)

Database Management System (DBMS)-এর মূল উদ্দেশ্য হলো Database-এ সুবিধাজনক ও দক্ষতার সাথে information সংরক্ষণ ও Database থেকে information পুনরুদ্ধার (retriving) করা। অন্যভাবে বলা যায়, DBMS হলো এমন এক ধরনের File system, যেখানে information-কে Structuring করার কাজ সম্পন্ন করা হয়, যেমন- Index sequential file organization। বেশির ভাগ প্রোগ্রামার Information management-এর এ লেভেলে কাজ করেন।

ফাইল সিস্টেমঃ যে পদ্ধতিতে file সংরক্ষণ ও file organize করা যায়, তাকে file system বলা হয়। File system-এর মাধ্যমে file-এ সংরক্ষিত data সহজে খুঁজে বের করা যায় এবং প্রয়োজন অনুসারে access করা যায়। File system, প্রয়োজনে বিভিন্ন ধরনের Data storage device যেমন- Hard disk বা CD-ROM ব্যবহার করতে পারে। File system-এ data'র logical organization থাকে এবং information management নিয়ে আলোচনা করা হয়।

অন্য কথায়, File system হলো Data storage, Organization, Manipulation এবং Data retrieval-এর জন্য একটি বিশেষ ধরনের Database যা O/S কর্তৃক নিয়ন্ত্রিত।

২। ফাইল সিস্টেমের সংগঠন বর্ণনা কর।

উত্তরঃ-

(i) অ্যাক্সেস মেথড (Access Method) : এ পর্যায়ে File-এর Logical record নাম্বারকে Logical byte address-এ রূপান্তর করে। এরপর Control-কে Logical file system-এ পাঠানো হয়।

(ii) লজিক্যাল ফাইল সিস্টেম (Logical file system) : এ পর্যায়ে Symbolic file নামকে-এর অনুরূপ (Corresponding) VTCC (Volume table of content) এন্ট্রি নম্বরে রূপান্তর করে। যদি Active Name Table (ANT)-এ নাম পাওয়া না যায়, তবে Master directory ও Subdirectory-এর মাধ্যমে খুঁজে বের করা হয়।

(iii) বেসিক ফাইল সিস্টেম (Basic file system) : যদি একটি file আগে থেকেই active file এ খোলা থাকে, তবে এ ক্ষেত্রে VTOC এন্ট্রি নাম্বার ব্যবহার করা হয়। আর যদি file খোলা না থাকে তাহলে Active file table-এ Copy করা হয়।

(iv) ফাইল অর্গানাইজেশন কৌশল মডিউল (File organization strategy module: FOSM) : এ ধাপে Logical byte address টি Logical number ও Offset-এ রূপান্তরিত হয়। File map ব্যবহারের মাধ্যমে Logical block number টি Physical block number-এ রূপান্তর হয়।

যদি আকাঙ্ক্ষিত block টি আগে থেকেই buffer-এ অবস্থান করে তবে Logical record-সমূহ Caller-কে ফেরত পাঠানো হয়। যদি আকাঙ্ক্ষিত block টি buffer এ না থাকে, তবে Device strategy module টি Physical block-কে call করে এবং read করে। read করার পর Logical record সমূহে Caller-কে ফেরত পাঠানো হয়। আবার যদি write request হয়, তবে অনুরূপ Physical block টি Allocation strategy module দ্বারা assign করা হয়।

(v) অ্যালোকেশন সিস্টেম কৌশল মডিউল (Allocation system strategy module: ASM) : এক্ষেত্রে একটি মুক্ত Physical block-কে মুক্ত file হতে অপসারিত করা হয় এবং পরে তা মনোনীত ফাইলের ফাইল ম্যাপ (Map)-এ স্থাপন করা হয়।

(vi) ডিভাইস কৌশল মডিউল (Device strategy module: DSM) : এ পর্যায়ে Physical block নাম্বার Address format-এ রূপান্তরিত হয়। Address format বলতে Device-এর সিলিভার, নাম্বার, ট্যাক নাম্বার ও Physical record নাম্বার ইত্যাদিকে বুঝানো হয়। এরপর প্রয়োজনীয় I/O Command স্থাপন এবং তারপর নিয়ন্ত্রণ I/O সিস্টেমে গমন করে।

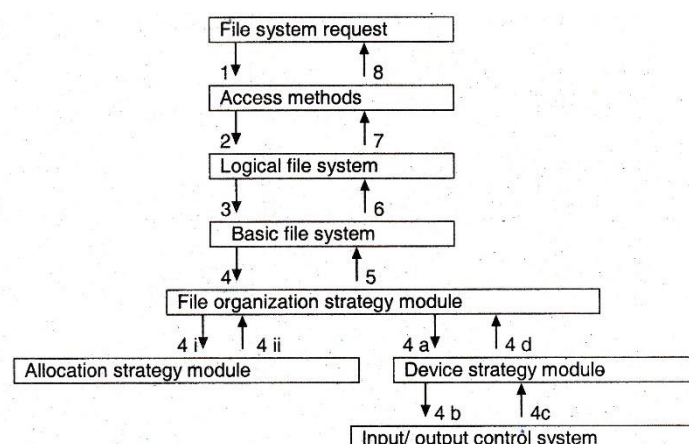
(vii) ইনপুট/আউটপুট কন্ট্রোল সিস্টেম (I/O control systems: IOCS) : এ পর্যায়ে I/O channel ব্যবহার করার Schedule করা, I/O Command-কে Initialization করা এবং Interrupt procedure প্রসেস করা হয়। যখন I/O অপারেশন শেষ হয়, তখন নিয়ন্ত্রণ Caller-কে ফেরত পাঠায়।

রচনামূলক প্রশ্ন

১। চিত্রসহ File System Organization বর্ণনা কর।

উত্তরঃ- একটি বহুধাপিক প্রক্রিয়া প্রত্যেকটি ধাপই যথেষ্ট গুরুত্বপূর্ণ। সাধারণত -কে অনুযায়ী সাতটি ধাপে ভাগ করা যায়। একটি করার জন্য একজন নামের সাহায্যে একটি পাঠায়। উক্ত নাম গ্রহণ করবে এবং তার অনুরূপ নিউমেরিক খুঁজে বের করবে। উক্ত খুঁজে বের করবে। নিউমেরিক গ্রহণ করবে এবং একটি পাবে। -এর ফাইল ব্যবহার করে -এর মান বের করবে। ইনফরমেশন করার জন্য তৈরি করবে এবং পরিশেষে -এর জন্য নির্বাহের সিডিউল তৈরি করবে। -কে অনুযায়ী যে সাতটি ধাপে ভাগ করা যায়, তা নিম্নে দেয়া হলোঃ

- (i) অ্যাক্সেস মেথড (Access Method)
- (ii) লজিক্যাল ফাইল সিস্টেম (Logical file system)
- (iii) বেসিক ফাইল সিস্টেম (Basic file system)
- (iv) ফাইল অর্গানাইজেশন কৌশল মডিউল (File organization strategy module: FOSM)
- (v) অ্যালোকেশন সিস্টেম কৌশল মডিউল (Allocation system strategy module: ASM)
- (vi) ডিভাইস কৌশল মডিউল (Device strategy module: DSM)
- (vii) ইনপুট/আউটপুট কন্ট্রোল সিস্টেম (I/O control systems: IOCS)



(i) **অ্যাক্সেস মেথড (Access Method)** : এ পর্যায়ে File-এর Logical record নাম্বারকে Logical byte address-এ রূপান্তর করে। এরপর Control-কে Logical file system-এ পাঠানো হয়।

(ii) **লজিক্যাল ফাইল সিস্টেম (Logical file system)** : এ পর্যায়ে Symbolic file নামকে-এর অনুরূপ (Corresponding) VTCC (Volume table of content) এন্ট্রি নম্বরে রূপান্তর করে। যদি Active Name Table (ANT)-এ নাম পাওয়া না যায়, তবে Master directory ও Subdirectory-এর মাধ্যমে খুঁজে বের করা হয়।

(iii) **বেসিক ফাইল সিস্টেম (Basic file system)** : যদি একটি file আগে থেকেই active file এ খোলা থাকে, তবে এ ক্ষেত্রে VTOC এন্ট্রি নাম্বার ব্যবহার করা হয়। আর যদি file খোলা না থাকে তাহলে Active file table-এ Copy করা হয়।

(iv) **ফাইল অর্গানাইজেশন কৌশল মডিউল (File organization strategy module: FOSM)** : এ ধাপে Logical byte address টি Logical number ও Offset-এ রূপান্তরিত হয়। File map ব্যবহারের মাধ্যমে Logical block number টি Physical block number-এ রূপান্তর হয়।

যদি আকাঙ্ক্ষিত block টি আগে থেকেই buffer-এ অবস্থান করে তবে Logical record-সমূহ Caller-কে ফেরত পাঠানো হয়। যদি আকাঙ্ক্ষিত block টি buffer এ না থাকে, তবে Device strategy module টি Physical block-কে call করে এবং read করে। read করার পর Logical record সমূহে Caller-কে ফেরত পাঠানো হয়। আবার যদি write request হয়, তবে অনুরূপ Physical block টি Allocation strategy module দ্বারা assign করা হয়।

(v) **অ্যালোকেশন সিস্টেম কৌশল মডিউল (Allocation system strategy module: ASM)** : এক্ষেত্রে একটি মুক্ত Physical block-কে মুক্ত file হতে অপসারিত করা হয় এবং পরে তা মনোনীত ফাইলের ফাইল ম্যাপ (Map)-এ স্থাপন করা হয়।

(vi) **ডিভাইস কৌশল মডিউল (Device strategy module: DSM)** : এ পর্যায়ে Physical block নাম্বার Address format-এ রূপান্তরিত হয়। Address format বলতে Device-এর সিলিন্ডার, নাম্বার, ট্যাক নাম্বার ও Physical record নাম্বার ইত্যাদিকে বুঝানো হয়। এরপর প্রয়োজনীয় I/O Command স্থাপন এবং তারপর নিয়ন্ত্রণ I/O সিস্টেমে গমন করে।

(vii) **ইনপুট/আউটপুট কন্ট্রোল সিস্টেম (I/O control systems: IOCS)** : এ পর্যায়ে I/O channel ব্যবহার করার Schedule করা, I/O Command-কে Initialization করা এবং Interrupt procedure প্রসেস করা হয়। যখন I/O অপারেশন শেষ হয়, তখন নিয়ন্ত্রণ Caller-কে ফেরত পাঠায়।

২। Disk-এর Space Management পদ্ধতি বর্ণনা কর।

উত্তরঃ- যেহেতু disk-এর space সীমিত, তাই disk-এর পুরো space-এর সদ্যবহার অতীব জরুরি। এজন্য disk-এর Free space ও Located space সম্পর্কে জানা আবশ্যিক। Disk হতে কোনো file-কে delete করে ফেললে ঐ Free space-এ নতুন কোনো file বা data সংরক্ষণ করা যায়। তাই Disk-এর কোন অংশটি free, তা শনাক্ত করে এর সঠিক ব্যবহার করাই হচ্ছে Free space management। সাধারণত চারটি পদ্ধতি প্রয়োগ করে Disk-রে Free space-কে শনাক্ত করা যায়, যেমন-

(i) **বিট ভেক্টর (Bit Vector)**

(ii) **লিঙ্কড লিস্ট (Linked List)**

(iii) **গ্রুপিং (Grouping)**

(vi) **কাউন্টিং (Counting)**

নিম্নে **Free space** শনাক্ত করার বিভিন্ন পদ্ধতিগুলো আলোচনা করা হলো:

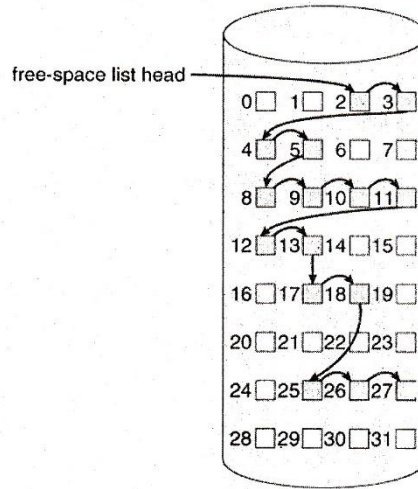
বিট ভেক্টর (Bit Vector) : এ পদ্ধতিতে disk-এর Free space list-কে bit map বা bit vector দ্বারা নির্ধারণ করা হয়। যদি disk-এর কোনো Block free হয়, তবে তাকে 1; আবার যদি block টি allocated হয়, তবে তাকে 0 (Zero) দ্বারা নির্দিষ্ট করা হয়।

উদাহরণঃ যদি কোনো disk-এর 1, 3, 5, 7, 9, 11 block-সমূহ free ও 2, 4, 6, 8, 10, 12 block-সমূহ allocated হয়, তবে এর Bit Map বা Bit Vector হবে নিম্নরূপঃ

0101010101010

Bit map বা bit vector পদ্ধতি তখনই বেশি প্রয়োজন হয়, যখন যথেষ্ট পরিমাণে Main memory available থাকে। এ পদ্ধতির সুবিধা হচ্ছে এটি বেশ সহজ পদ্ধতি, ভালোভাবে অনুধাবন করা যায় এবং দ্রুত free block খুঁজে বের করা যায়। এটি একটি দক্ষ পদ্ধতি।

লিঙ্কড লিস্ট (Linked List) : এ পদ্ধতিতে free block-গুলোকে একত্রে Link করা হয় এবং একটি Pointer-কে disk-এর বিশেষ অবস্থানে free block-এর শুরুতে রাখা হয়। পরবর্তীতে ২য় free block-এর সাথে ১ম free block-এর pointer-এর link, ৩য় free block-এর সাথে ২য় free block-এর Pointer linked হয়। এরূপ ধারাটি চলতে থাকে, যা নিম্নের চিত্রে দেখানো হলো।



গ্রুপিং (Grouping) : এ পদ্ধতিতে disk-এর প্রথম free block-এ সংখ্যক free block-এর address জমা করে রাখে। এতে প্রথম (n-1) সংখ্যক block সাধারণত free থাকে। সর্বশেষ block টি অন্য n সংখ্যক free block-এর address ধারণ করে এবং এরূপ প্রক্রিয়াটি চলতে থাকে।

এ পদ্ধতির প্রধান সুবিধা হচ্ছে একসঙ্গে অনেকগুলো free block-এর address খুঁজে পাওয়া যায়।

কাউন্টিং (Counting) : এ পদ্ধতিতে পরস্পর সংলগ্ন কতগুলো Block allocated হয় কিংবা free হয়। এরূপেই প্রক্রিয়াটি চলতে থাকে। সেজন্য প্রথম Free block address ও কতগুলো Block free তা Count করে রাখতে হয়। এভাবে প্রতিটি entry-তেই n-সংখ্যক থাকে Free block ও তাদের জন্য address-ও থাকে।