Empowerment Through Quality Technical Education
AJEENKYA DY PATIL SCHOOL OF ENGINEERING
Dr. D. Y. Patil Knowledge City, Charholi (Bk), Lohegaon, Pune – 412 105
Website: https://dypsoe.in/

LAB MANUAL

Laboratory Practice-I (310248)
TE (COMP) 2019 COURSE

Course Coordinator
Prof. Priti B. Rathod
Prof. Sangeeta Mohapatra

Prof. Pooja Mohbansi

DEPARTMENT OF
COMPUTER ENGINEERING

Department of Computer Engineering

Vision: "To achieve excellence in technical and socio-economic fields"

Mission:

M1 : To develop excellent learning center through continuous up gradation in proximity with Academia . R&D centers and industries.

M2 : To pursue research of local and global relevance.

M3: To encourage students to consider "startups" as a career option through Entrepreneurship Development Cell.

M4: Uplift and groom the learners to emerge as committed professionals.

.

Program Educational Outcomes:

Our graduates will be able to,

PEO1: be globally competent having strong fundamentals, domain knowledge, updated with modern technology to provide the effective solutions for engineering problems.

PEO2: Work as a committed professional v.ith strong professional ethics and values. sense of responsibilities. Understanding of legal. sulcty. hculth. societal. cultural and environmental issues.

PEO3: be committed and motivated graduates with research attitude lifelong learning, investigative approach and multidisciplinary thinking.

Program Specific Outcomes:

PSO1	**Professional Skills**-The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexities.
PSO2	**Problem-Solving Skills**- The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PSO3	Successful Career and Entrepreneurship- The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur and to have a zest for higher studies.

Table of Contents

Contents

1. Guidelines to manual usage

This manual assumes that the facilitators are aware of collaborative learning methodologies.

This manual will provide a tool to facilitate the session on Digital Communication modules in collaborative learning environment.

The facilitator is expected to refer this manual before the session.

Icon of Graduate Attributes

K Applying Knowledge	**A** Problem Analysis	**D** Design & Development	**I** Investigation of problems
M Modern Tool Usage	**E** Engineer & Society	**E** Environment Sustainability	**T** Ethics
T Individual & Team work	**O** Communication	**M** Project Management & Finance	**I** Life-Long Learning

Disk Approach- Digital Blooms Taxonomy

1: Remembering / Knowledge

2: Comprehension / Understanding

3: Applying

4: Analyzing

5: Evaluating

6: Creating / Design

Program Outcomes:

PO1	**Engineering knowledge**	Apply the knowledge of mathematics, science, Engineering fundamentals, and an Engineering specialization to the solution of complex Engineering problems.
PO2	**Problem analysis**	Identify, formulate, review research literature and analyze complex Engineering problems reaching substantiated conclusions using first principles of Mathematics, natural sciences and Engineering sciences.
PO3	**Design / Development of Solutions**	Design solutions for complex Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and Environmental considerations.
PO4	**Conduct Investigations of Complex Problems**	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern Tool Usage**	Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modeling to complex Engineering activities with an understanding of the limitations.
PO6	**The Engineer and Society**	Apply reasoning informed by the contextual knowledge to assess societal, health, Safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and Sustainability**	Understand the impact of the professional Engineering solutions in societal and Environmental contexts, and demonstrate the knowledge of, and need for Sustainable development.
PO8	**Ethics**	Apply ethical principles and commit to professional ethics and responsibilities And norms of Engineering practice.
PO9	**Individual and Team Work**	Function effectively as an individual, and as a member or leader in diverse Teams, and in multidisciplinary settings.
PO10	**Communication Skills**	Communicate effectively on complex Engineering activities with the Engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make Effective presentations, and give and receive clear instructions.
PO11	**Project Management and Finance**	Demonstrate knowledge and understanding of Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary Environments.
PO12	**Life-long Learning**	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological Change.

Course Name: Laboratory Practice-I

Course Code: 310248

Course Outcomes

Course Name: Laboratory Practice-II

Course Code: 310258

Course Outcomes

On completion of the course, learners will be able to

Systems Programming and Operating System

- **CO1:** Implement language translators
- **CO2**: Use tools like LEX and YACC
- **CO3**: Implement internals and functionalities of Operating System

Software Project Management

- CO4:Apply Software Project Management tools
- CO5:Implement software project planning and scheduling
- CO6:Analyse staffing in software project

CO to PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	-	2	-	3	-	-	2	2	2	1	2
CO2	1	-	2	2	3	2	-	2	2	2	1	2
CO3	1	-	2	2	3	2	-	2	2	2	2	2
CO4	1	-	2	-	3	-	-	2	2	2	2	2
CO5	1	-	2	-	3	-	-	2	2	2	2	2
CO6	1	-	2	-	3	-	-	2	2	2	2	2

CO to PSO Mapping:

	PSO1	PSO2	PSO3
CO1		2	
CO2		3	1
CO3	**2**	**3**	
CO4	2	2	
CO5	**1**	**3**	2
CO6	3		

1. **Laboratory Objective**

- To learn system programming tools
- To learn modern operating system
- To learn various techniques, tools, applications in IoT and Embedded Systems /Human Computer Interface/Distributed Systems/ Software Project Management

2. Laboratory Equipment/Software

Operating System recommended: - 64-bit Windows OS and Linux **Programming tools recommended: -**

Software:- Eclips Java

Backend: MySQL /MongoDB/NodeJS

Software project management-MS project/Gantt Project/Primavera

3. Laboratory Experiment list

Sr. No	Title
	Prerequisite practical assignments or installation (if any)
1	Eclips Java
	Part I : Systems Programming and Operating System
	Group A (Any Two Assignments from Sr. No. 1 to 3)
1	Design suitable Data structures and implement Pass-I and Pass-II of a two-pass assembler for pseudo-machine. Implementation should consist of a few instructions from each category and few assembler directives. The output of Pass-I (intermediate code file and symbol table) should be input for Pass-II.
2	Write a program to create Dynamic Link Library for any mathematical operation and write an application program to test it. (Java Native Interface / Use VB or VC++).
4	Write a program to simulate CPU Scheduling Algorithms: FCFS, SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive).**Group B**
5	Write a program to simulate Page replacement algorithm.
	Part II : Elective I
	Software Project Management
6	**Create Project Plan** ▪ Specify project name and start (or finish) date. ▪ Identify and define project tasks. ▪ Define duration for each project task. ▪ Define milestones in the plan ▪ Define dependency between tasks ▪ Define project calendar. ▪ Define project resources and specify resource type Assign resources against each task and baseline the project plan
7	**Execute and Monitor Project Plan** ▪ Update % Complete with current task status. ▪ Review the status of each task. ▪ Compare Planned vs Actual Status ▪ Review the status of Critical Path Review resources assignation status
8	**Generate Dashboard and Reports** • **Dashboard** o Project Overview o Cost Overview o Upcoming Tasks • **Resource Reports** o Over-allocated Resources o Resource Overview • **Cost Reports** o Earned Value Report o Resource Cost Overview o Task Cost Overview • **Progress Reports** o Critical Tasks o Milestone Report

	o Slipping Tasks

GROUP - A

EXPERIMENT NO : 01

1. Title:

Design suitable Data structures and implement Pass-I and Pass-II of a two-pass assembler for pseudo-machine. Implementation should consist of a few instructions from each category and few assembler directives. The output of Pass-I (intermediate code file and symbol table) should be input for Pass-II..

2. Objectives :

- To understand Data structure of Pass-1 assembler
- To understand Pass-1 assembler concept
- To understand Advanced Assembler Directives

3. Problem Statement :

Design suitable data structures and implement Pass-I and Pass-II of a two-pass assembler for pseudo-machine in Java using object oriented feature.

4. Outcomes:

After completion of this assignment students will be able to:

- Implemented Pass-I and Pass-II assembler
- Implemented Symbol table, Literal table & Pool table, Intermediate Code, Machine code
- Understood concept Advanced Assembler Directive.

5. Software Requirements:

Latest jdk, Eclipse

6. Hardware Requirement:

- M/C Lenovo Think center M700 Ci3,6100,6th Gen. H81, 4GB RAM ,500GB HDD

7. Theory Concepts:

Introduction :-

There are two main classes of programming languages: *high level* (e.g., C, Pascal) and *low level*. *Assembly Language* is a low level programming language. Programmers code symbolic instructions, each of which generates machine instructio

An *assembler* is a program that accepts as input an assembly language program (source) and produces its machine language equivalent (object code) along with the information for the loader.

Figure 1. Executable program generation from an assembly source code

Advantages of coding in assembly language are:

- Provides more control over handling particular hardware components
- May generate smaller, more compact executable modules
- Often results in faster execution

Disadvantages:

- Not portable
- More complex
- Requires understanding of hardware details (interfaces)

Pass – 1 Assembler:

An assembler does the following:

1. Generate machine instructions
    - evaluate the mnemonics to produce their machine code
    - evaluate the symbols, literals, addresses to produce their equivalent machine addresses
    - convert the data constants into their machine representations
2. Process pseudo operations

Pass – 2 Assembler:

A two-pass assembler performs two sequential scans over the source code:

Pass 1: symbols and literals are defined
Pass 2: object program is generated

Parsing: moving in program lines to pull out op-codes and operands

Data Structures:

- **Location counter (LC):** points to the next location where the code will be placed
- **Op-code translation table**: contains symbolic instructions, their lengths and their op-codes (or subroutine to use for translation)
- **Symbol table (ST):** contains labels and their values
- **String storage buffer (SSB):** contains ASCII characters for the strings
- **Forward references table (FRT):** contains pointer to the string in SSB and offset where its value will be inserted in the object code

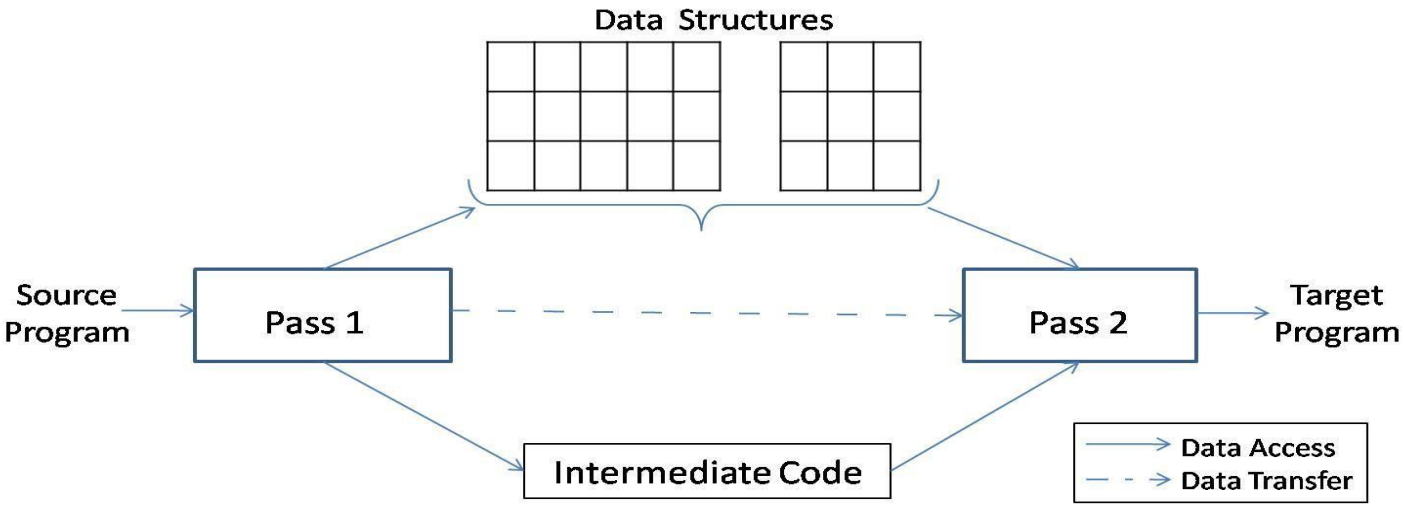


Figure 2. ***A simple two pass assembler.***

Elements of Assembly Language :

An assembly language programming provides three basic features which simplify programming when compared to machine language.

1. **Mnemonic Operation Codes :**

Mnemonic operation code / Mnemonic Opcodes for machine instruction eliminates the need to memorize numeric operation codes. It enables assembler to provide helpful error diagnostics. Such as indication of misspelt operation codes.

2. **Symbolic Operands :**

Symbolic names can be associated with data or instructions. These symbolic names can be used as operands in assembly statements. The assembeler performes memory bindinding to these names; the programmer need not know any details of the memory bindings performed by the assembler.

3. **Data declarations :**

Data can be declared in a variety of notations, including the decimal notation. This avoids manual conversion of constants into their internal machine representation, for example -5 into (11111010)2 or 10.5 into (41A80000)16

Statement format :

An assembly language statement has the following format :

[Label] <Opcode> <operand Spec> [, operand Spec> ..]

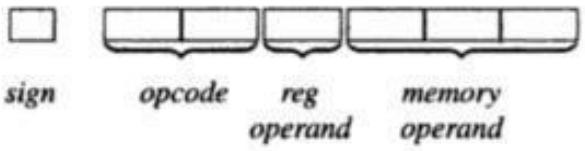
Where the notation [..] indicates that the enclosed specification is optional.

Label associated as a symbolic name with the memory word(s) generated for the statement

Mnemonic Operation Codes :

Instruction opcode	*Assembly mnemonic*	*Remarks*
00	STOP	Stop execution
01	ADD	*First operand is modified* *Condition code is set*
02	SUB	
03	MULT	
04	MOVER	Register ← memory move
05	MOVEM	Memory ← register move
06	COMP	Sets condition code
07	BC	Branch on condition
08	DIV	Analogous to SUB
09	READ	*First operand is not used*
10	PRINT	

Instruction Format :



Sign is not a part of Instruction

An Assembly and equivalent machine language program :(solve it properly)

```
          START   101
          READ    N                 101)  + 09 0 113
          MOVER   BREG, ONE         102)  + 04 2 115
          MOVEM   BREG, TERM        103)  + 05 2 116
AGAIN     MULT    BREG, TERM        104)  + 03 2 116
          MOVER   CREG, TERM        105)  + 04 3 116
          ADD     CREG, ONE         106)  + 01 3 115
          MOVEM   CREG, TERM        107)  + 05 3 116
          COMP    CREG, N           108)  + 06 3 113
          BC      LE, AGAIN         109)  + 07 2 104
          MOVEM   BREG, RESULT      110)  + 05 2 114
          PRINT   RESULT            111)  + 10 0 114
          STOP                      112)  + 00 0 000
N         DS      1                 113)
RESULT    DS      1                 114)
ONE       DC      '1'               115)  + 00 0 001
TERM      DS      1                 116)
          END
```

Note : you can also take other example with solution

Assembly Language Statements :

Three Kinds of Statements

1. Imperative Statements
2. Declaration Statements
3. Assembler Directives

a) **Imperative Statements :** It indicates an action to be performed during the execution of the assembled program. Each imperative statement typically translates into one machine instruction.
e.g, MOVER, ADD, MULT etc (All executable statements)

b) **Declaration Statements :** Two types of declaration statements is as follows

[Label] DS <constant>
[Label] DC '<Value>'

The DS (Declare Storage) statement reserves areas of memory and associates names with them.

Eg)**A DS 1**
B DS 150

First statement reserves a memory of 1 word and associates the name of the memory as A.
Second statement reserves a memory of 150 word and associates the name of the memory as B.

The DC (Declare Constant) Statement constructs memory word containing constants.

Eg) **ONE DC '1'**

Associates the name ONE with a memory word containing the value '1' . The programmer can declare constants in decimal, binary, hexadecimal forms etc., These values are not protected by the assembler. In the above assembly language program the value of ONE Can be changed by executing an instruction MOVEM BREG,ONE

c. **Assembler Directives :**

Assembler directives instruct the assembler to perform certain actions during the assembly of a program. Some Assembler directives are described in the following

START <Constant>

Indicates that the first word of the target program generated by the assembler should be placed in the memory word with address <Constant>

END [<operand spec>]

It Indicates the end of the source program

Pass Structure of Assembler :

One complete scan of the source program is known as a pass of a Language Processor.

Two types 1) Single Pass Assembler 2) Two Pass Assembler.

Single Pass Assembler :

First type to be developed Most Primitive Source code is processed only once.

The operand field of an instruction containing forward reference is left blank intially

Eg) MOVER BREG,ONE

Can be only partially synthesized since ONE is a forward reference

During the scan of the source program, all the symbols will be stored in a table called **SYMBOL TABLE.** Symbol table consists of two important fields, they are symbol name and address.

All the statements describing forward references will be stored in a table called Table of Incompleted Instructions (TII)

TII (Table of Incomplete instructions)

Instruction Address	Symbol
101	ONE

By the time the END statement is processed the symbol table would contain the address of all symbols defined in the source program.

Two Pass Assembler :

Can handle forward reference problem easily.

First Phase : *(Analysis)*

- Symbols are entered in the table called Symbol table
- Mnemonics and the corresponding opcodes are stored in a table called Mnemonic table
- LC Processing

Second Phase : *(Synthesis)*

- Synthesis the target form using the address information found in Symbol table.
- First pass constructs an Intermediated Representation (IR) of the source program for use by the second pass.

Data Structure used during Synthesis Phase :

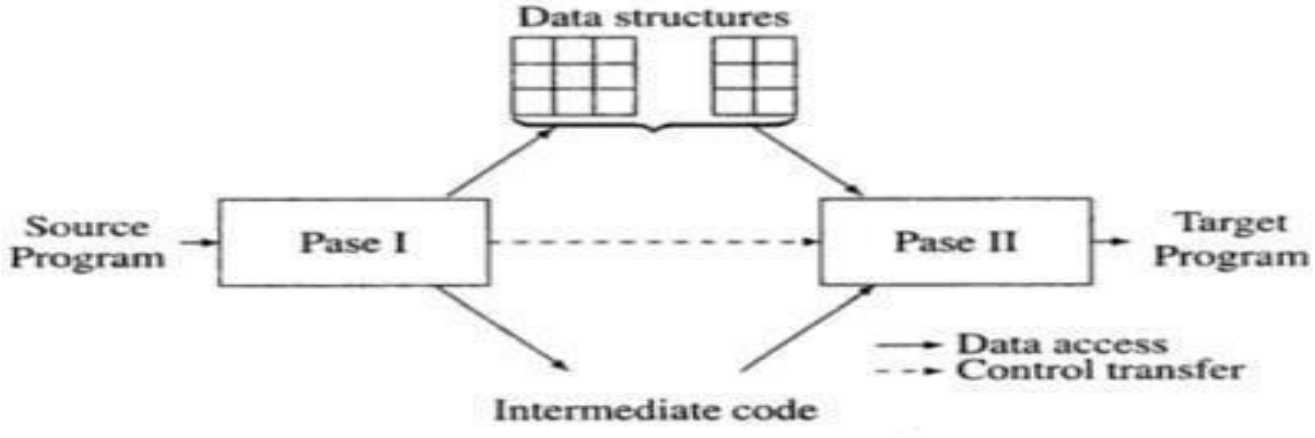
1. Symbol table
2. Mnemonics table

symbol	*address*
AGAIN	104
N	113

mnem-onic	*op code*	*length*
ADD	01	1
SUB	02	1

mnem-onic	*op code*	*length*
ADD	01	1
SUB	02	1

Processed form of the source program called Intermediate Code (IC)



ADVANCED ASSEMBLER DIRECTIVES

1. ORIGIN
2. EQU
3. LTROG

ORIGIN :

Syntax : ORIGIN < address spec>

< address spec>can be an <operand spec> or constant

Indicates that Location counter should be set to the address given by < address spec>

This statement is useful when the target program does not consist of consecutive memory words.

Eg) ORIGIN Loop + 2

EQU :

Syntax

<symbol> EQU <address spec>

<address spec>operand spec (or) constant
Simply associates the name symbol with address specification No

Location counter processing is implied

Eg) Back EQU Loop

LTORG : (Literal Origin)

Where should the assembler place literals ?

It should be placed such that the control never reaches it during the execution of a program.

By default, the assembler places the literals after the END statement.

LTROG statement permits a programmer to specify where literals should be placed.

Note :(you can also write your own theory for this practical)

Solve the below example.for Pass-1 & Pass-2 of Two Pass Assembler.

```
          START 200
          MOVER AREG, ='5'
          MOVEM AREG, X
L1        MOVER BREG, ='2'
          ORIGIN L1+3
          LTORG
NEXT      ADD AREG, ='1'
          SUB BREG, ='2'
          BC LT, BACK
```

Thus , We have implemented Pass-1 & Pass-2 assembler with symbol table, literal table and pool table, Intermediate code and Machine code.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

– TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code, PN - Punctuality and Neatness.

GROUP - A

EXPERIMENT NO : 02

1. Title:

Write a program to create Dynamic Link Library for any mathematical operation and write an application program to test it. (Java Native Interface / Use VB or VC++).

2. Objectives :

- To understand Dynamic Link Libraries Concepts
- To implement dynamic link library concepts
- To study about Visual Basic

3. Problem Statement :

Write a program to create Dynamic Link Library for Arithmetic Operation in VB.net

4. Outcomes:

After completion of this assignment students will be able to:

- Understand the concept of Dynamic Link Library
- Understand the Programming language of Visual basic

5. Software Requirements:

- Visual Studio 2010

6. Hardware Requirement:

- M/C Lenovo Think center M700 Ci3,6100,6th Gen. H81, 4GB RAM ,500GB HDD

7. Theory Concepts:

Dynamic Link Library :

A dynamic link library (DLL) is a collection of small programs that can be loaded when needed by larger programs and used at the same time. The small program lets the larger program communicate with a specific device, such as a printer or scanner. It is often packaged as a DLL program, which is usually referred to as a DLL file. DLL files that support specific device operation are known as device drivers.

A DLL file is often given a ".dll" file name suffix. DLL files are dynamically linked with the program that uses them during program execution rather than being compiled into the main program.

The advantage of DLL files is space is saved in random access memory (RAM) because the files don't get loaded into RAM together with the main program. When a DLL file is needed, it is loaded and run. For example, as long as a user is editing a document in Microsoft Word, the printer DLL file does not need to be loaded into RAM. If the user decides to print the document, the Word application causes the printer DLL file to be loaded and run.

A program is separated into modules when using a DLL. With modularized components, a program can be sold by module, have faster load times and be updated without altering other parts of the program. DLLs help operating systems and programs run faster, use memory efficiently and take up less disk space.

Feature of DLL :

DLLs are essentially the same as EXEs, the choice of which to produce as part of the linking process is for clarity, since it is possible to export functions and data from either.

- It is not possible to directly execute a DLL, since it requires an EXE for the operating system to load it through an entry point, hence the existence of utilities like RUNDLL.EXE or RUNDLL32.EXE which provide the entry point and minimal framework for DLLs that contain enough functionality to execute without much support.

- DLLs provide a mechanism for shared code and data, allowing a developer of shared code/data to upgrade functionality without requiring applications to be re-linked or re-compiled. From the application development point of view Windows and OS/2 can be thought of as a collection of DLLs that are upgraded, allowing applications for one version of the OS to work in a later one, provided that the OS vendor has ensured that the interfaces and functionality are compatible.

- DLLs execute in the memory space of the calling process and with the same access permissions which means there is little overhead in their use but also that there is no protection for the calling EXE if the DLL has any sort of bug.

Difference between the Application & DLL :

- An application can have multiple instances of itself running in the system simultaneously, whereas a DLL can have only one instance.
- An application can own things such as a stack, global memory, file handles, and a message queue, but a DLL cannot.

Executable file links to DLL :

An executable file links to (or loads) a DLL in one of two ways:

- Implicit linking
- Explicit linking

Implicit linking is sometimes referred to as static load or load-time dynamic linking. Explicit linking is sometimes referred to as dynamic load or run-time dynamic linking.

With implicit linking, the executable using the DLL links to an import library (.lib file) provided by the maker of the DLL. The operating system loads the DLL when the executable using it is loaded. The client executable calls the DLL's exported functions just as if the functions were contained within the executable.

With explicit linking, the executable using the DLL must make function calls to explicitly load and unload the DLL and to access the DLL's exported functions. The client executable must call the exported functions through a function pointer.

An executable can use the same DLL with either linking method. Furthermore, these mechanisms are not mutually exclusive, as one executable can implicitly link to a DLL and another can attach to it explicitly.

DLL's Advantages :

- Saves memory and reduces swapping. Many processes can use a single DLL simultaneously, sharing a single copy of the DLL in memory. In contrast, Windows must load a copy of the library code into memory for each application that is built with a static link library.
- Saves disk space. Many applications can share a single copy of the DLL on disk. In contrast, each application built with a static link library has the library code linked into its executable image as a separate copy.
- Upgrades to the DLL are easier. When the functions in a DLL change, the applications that use them do not need to be recompiled or relinked as long as the function arguments and return values do not change. In contrast, statically linked object code requires that the application be relinked when the functions change.
- Provides after-market support. For example, a display driver DLL can be modified to support a display that was not available when the application was shipped.
- Supports multi language programs. Programs written in different programming languages can call the same DLL function as long as the programs follow the function's calling convention. The programs and the DLL function must be compatible in the following ways: the order in which the function expects its arguments to be pushed onto the stack, whether the function or the application is responsible for cleaning up the stack, and whether any arguments are passed in registers.

- Provides a mechanism to extend the MFC library classes. You can derive classes from the existing MFC classes and place them in an MFC extension DLL for use by MFC applications.
- Eases the creation of international versions. By placing resources in a DLL, it is much easier to create international versions of an application. You can place the strings for each language version of your application in a separate resource DLL and have the different language versions load the appropriate resources.

Disadvantage :

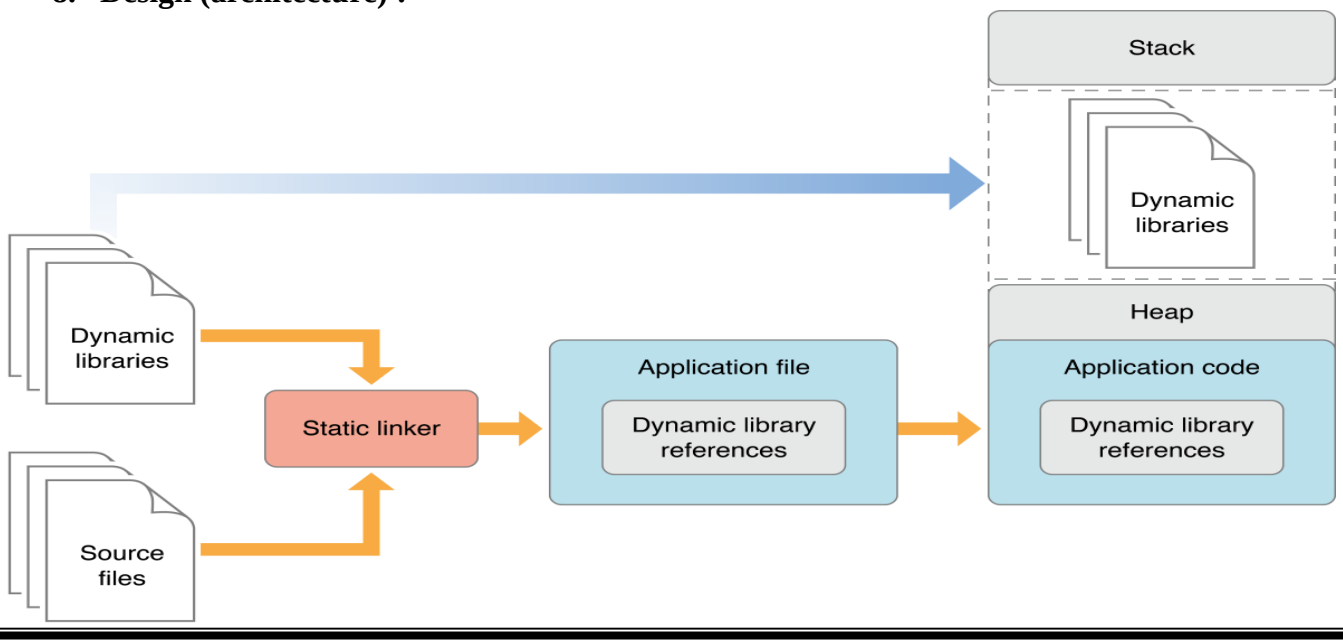
- A potential disadvantage to using DLLs is that the application is not self-contained; it depends on the existence of a separate DLL module.

Visual Basic :

Visual Basic is a third-generation event-driven programming language first released by Microsoft in 1991. It evolved from the earlier DOS version called BASIC. **BASIC** means **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. Since then Microsoft has released many versions of Visual Basic, from Visual Basic 1.0 to the final version Visual Basic 6.0. Visual Basic is a user-friendly programming language designed for beginners, and it enables anyone to develop GUI window applications easily.

In 2002, Microsoft released Visual Basic.NET(VB.NET) to replace Visual Basic 6. Thereafter, Microsoft declared VB6 a legacy programming language in 2008. Fortunately, Microsoft still provides some form of support for VB6. VB.NET is a fully object-oriented programming language implemented in the .NET Framework. It was created to cater for the development of the web as well as mobile applications. However, many developers still favor Visual Basic 6.0 over its successor Visual Basic.NET.

8. Design (architecture) :



STEP FOR DLL PROGRAM :

1. **Create new project** → project → visual basic → windows form application → name the project → click OK.

2. **Design form :**

3. Right click on solution → Add → new project → windows → empty project → name to dll file(p5dll) → OK.

4. Right click on **dll file** (p5dll) → add → module → renmame module name as p5dll → OK.

5. Right click on **dll file** → properties → application type (class library) → save file.

6. Right click on main project → add references (created dll file will be displayed) → select dll file → OK.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

– TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code, PN - Punctuality and Neatness.

GROUP - B

GROUP - B

EXPERIMENT NO : 03

1. **Title:**

   Write a program to solve classical problems of synchronization using mutex and semaphore.

2. **Objectives :**
   - To understand reader writer synchronization problem
   - To solve reader-writer synchronization problem using mutex and semaphore

3. **Problem Statement :**
   Write a program to solve classical problems of synchronization using mutex and emaphore. (Reader-Writer Problem)

4. **Outcomes:**

After completion of this assignment students will be able to:

- Understand the concept of Deadlock, Semaphore, Mutex
- Understand of Classical synchronization problem

5. **Software Requirements:**
   - Eclipse IDE

6. **Hardware Requirement:**

   - M/C Lenovo Think center M700 Ci3,6100,6th Gen. H81, 4GB RAM ,500GB HDD

7. **Theory Concepts:**
   There is a data area shared among a number of processor registers.

   • The data area could be a file, a block of main memory, or even a bank of processor registers.

   • There are a number of processes that only read the data area (readers) and a number that only write to the data area (writers).

   • The conditions that must be satisfied are

   ➢ Any number of readers may read simultaneously read the file.

   ➢ Only one write at a time may write to the file.

   ➢ If a writer is writing to the file, no reader may read it.

- **Classical Synchronization Problem :**

We will see number of classical problems of synchronization as examples of a large class of concurrency-control problems. In our solutions to the problems, we use semaphores for synchronization, since that is the traditional way to present such solutions. However, actual implementations of these solutions could use mutex locks in place of binary semaphores.

These problems are used for testing nearly every newly proposed synchronization scheme. The following problems of synchronization are considered as classical problems:

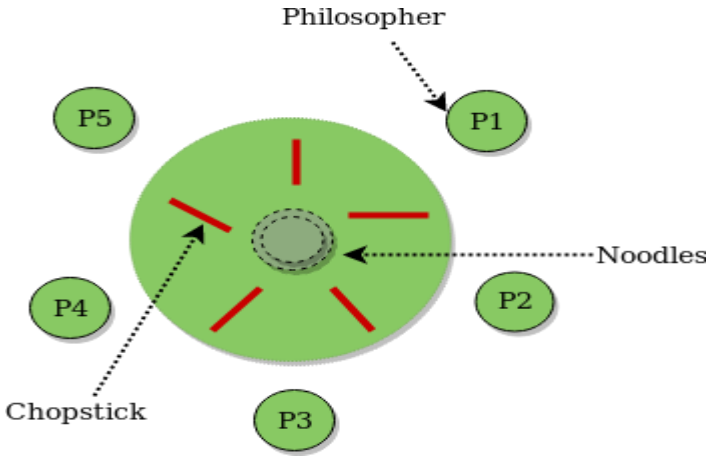
1. Bounded-buffer (or Producer-Consumer) Problem,
2. Dining-Philosophers Problem,
3. Readers and Writers Problem,
4. Sleeping Barber Problem

1. Bounded-buffer (or Producer-Consumer) Problem:

Bounded Buffer problem is also called producer consumer problem. This problem is generalized in terms of the Producer-Consumer problem. Solution to this problem is, creating two counting semaphores "full" and "empty" to keep track of the current number of full and empty buffers respectively. Producers produce a product and consumers consume the product, but both use of one of the containers each time.

2. Dining-Philosophers Problem:

The Dining Philosopher Problem states that K philosophers seated around a circular table with one chopstick between each pair of philosophers. There is one chopstick between each philosopher. A philosopher may eat if he can pickup the two chopsticks adjacent to him. One chopstick may be picked up by any one of its adjacent followers but not both. This problem involves the allocation of limited resources to a group of processes in a deadlock-free and starvation-free manner.



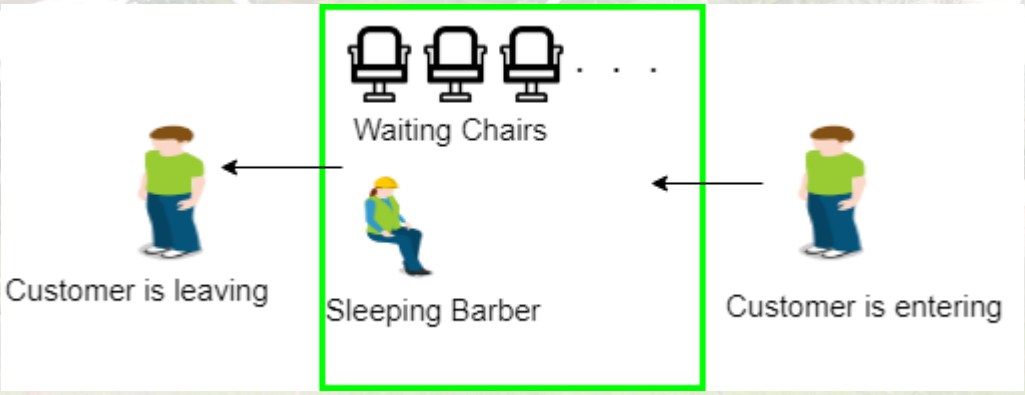
3. Readers and Writers Problem:

Suppose that a database is to be shared among several concurrent processes. Some of these processes may want only to read the database, whereas others may want to update (that is, to read and write) the database. We distinguish between these two types of processes by referring to the former as readers and to the latter as writers. Precisely in OS we call this situation as the readers-writers problem. Problem parameters:

- One set of data is shared among a number of processes.
- Once a writer is ready, it performs its write. Only one writer may write at a time.
- If a process is writing, no other process can read it.
- If at least one reader is reading, no other process can write.
- Readers may not write and only read.

4. Sleeping Barber Problem:

Barber shop with one barber, one barber chair and N chairs to wait in. When no customers the barber goes to sleep in barber chair and must be woken when a customer comes in. When barber is cutting hair new customers take empty seats to wait, or leave if no vacancy.



Semaphore:

Definition: Semaphores are system variables used for synchronization of process.

Semaphore can be used in other synchronization problems besides Mutual Exclusion.

Two types of Semaphore:

- ➢ **Counting semaphore –** integer value can range over an unrestricted domain
- ➢ **Binary semaphore –**

Integer value can range only between 0 and 1; can be simpler to implement

Also known as mutex locks

Semaphore functions:

Package: import java.util.concurrent.Semaphore;

1) To initialize a semaphore:

Semaphore Sem1 = new Semaphore(1);

2) To wait on a semaphore:

/* Wait (S)

while S<=0

no-op;

S - -;

*/

Sem1.acquire();

3) To signal on a semaphore:

/* Signal(S)

S ++;

*/ mutex.release();

8. Algorithms(procedure) :

Note: you should write algorithm & procedure as per program/concepts (its sample algo. For reference)

1. import java.util.concurrent.Semaphore;
2. Create a class RW
3. Declare semaphores – mutex and wrt
4. Declare integer variable readcount = 0
5. Create a nested class Reader implements Runnable

a. Override run method (Reader Logic)

i. wait(mutex);
ii. readcount := readcount +1;
iii. if readcount = 1 then
iv. wait(wrt);
v. signal(mutex);
vi. …
vii. reading is performed
viii. …
ix. wait(mutex);
x. readcount := readcount – 1;
xi. if readcount = 0 then signal(wrt);
xii. signal(mutex):

6. Create a nested class Writer implements Runnable

a. Override run method (Writer Logic)

i. wait(wrt);
ii. …

iii. writing is performed
iv. …
v. signal(wrt);
7. Create a class main
a. Create Threads for Reader and Writer
b. Start these thread

9. Flowchart :

Note: you should draw flowchart as per algorithm/procedure

10. Conclusion:

Thus, I have studied classical synchronization problem to implement reader-writer problem using semaphore and mutex.

GROUP - B

EXPERIMENT NO : 04

1. **Title:**

   Write a Java program (using OOP features) to implement following scheduling algorithms: FCFS , SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive).

2. **Objectives :**

   - To understand OS & SCHEDULLING Concepts
   - To implement Scheduling FCFS, SJF, RR & Priority algorithms
   - To study about Scheduling and scheduler

3. **Problem Statement :**

   Write a Java program (using OOP features) to implement following scheduling algorithms: FCFS , SJF, Priority and Round Robin .

4. **Outcomes:**

After completion of this assignment students will be able to:

- Knowledge Scheduling policies
- Compare different scheduling algorithms

5. **Software Requirements:**

   JDK/Eclipse

6. **Hardware Requirement:**

   - M/C Lenovo Think center M700 Ci3,6100,6th Gen. H81, 4GB RAM ,500GB HDD

7. **Theory Concepts:**

CPU Scheduling:

• CPU scheduling refers to a set of policies and mechanisms built into the operating systems that govern the order in which the work to be done by a computer system is completed.

• Scheduler is an OS module that selects the next job to be admitted into the system and next process to run.

• The primary objective of scheduling is to optimize system performance in accordance with the criteria deemed most important by the system designers.

What is scheduling?

Scheduling is defined as the process that governs the order in which the work is to be done. Scheduling is done in the areas where more no. of jobs or works are to be performed. Then it requires some plan i.e. scheduling that means how the jobs are to be performed i.e. order. CPU scheduling is best example of scheduling.

What is scheduler?

1. Scheduler in an OS module that selects the next job to be admitted into the system and the next process to run.
2. Primary objective of the scheduler is to optimize system performance in accordance with the criteria deemed by the system designers. In short, scheduler is that module of OS which schedules the programs in an efficient manner.

Necessity of scheduling

• Scheduling is required when no. of jobs are to be performed by CPU.

• Scheduling provides mechanism to give order to each work to be done.

• Primary objective of scheduling is to optimize system performance.

• Scheduling provides the ease to CPU to execute the processes in efficient manner.

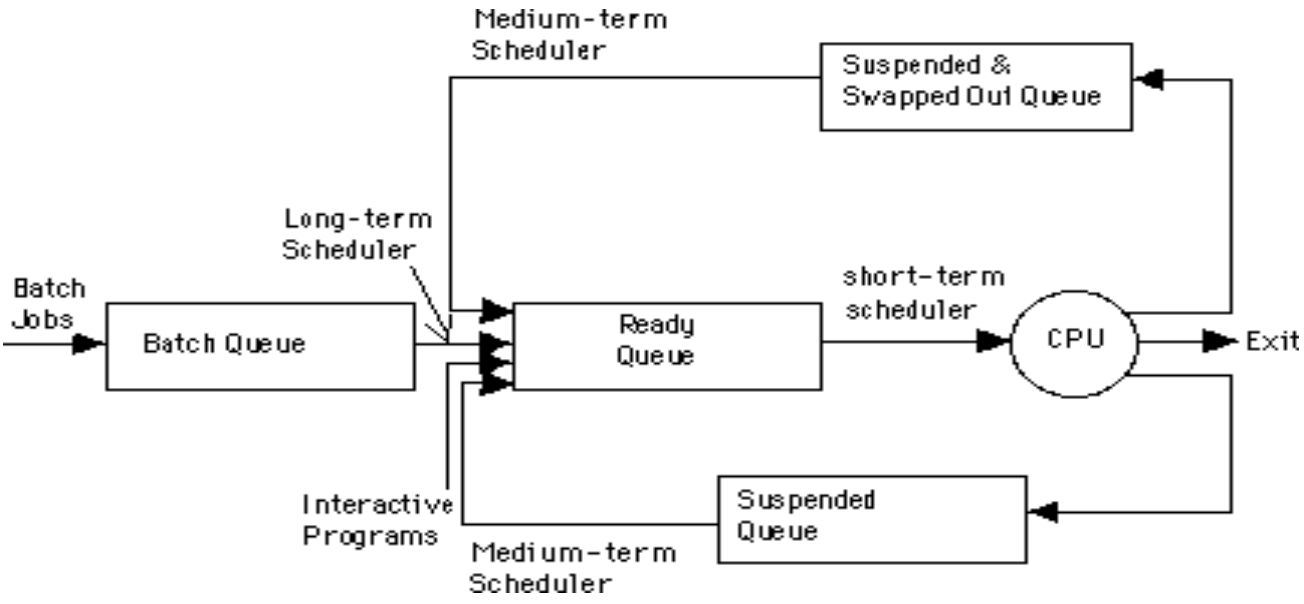
Types of schedulers

In general, there are three different types of schedulers which may co-exist in a complex operating system.

• Long term scheduler

• Medium term scheduler

• Short term scheduler.



Long Term Scheduler

• The long term scheduler, when present works with the batch queue and selects the next batch job to be executed.

• Batch is usually reserved for resource intensive (processor time, memory, special I/O devices) low priority programs that may be used fillers of low activity of interactive jobs.

• Batch jobs usually also contains programmer-assigned or system-assigned estimates of their resource needs such as memory size, expected execution time and device requirements.

• Primary goal of long term scheduler is to provide a balanced mix of jobs.

Medium Term Scheduler

• After executing for a while, a running process may because suspended by making an I/O request or by issuing a system call.

• When number of processes becomes suspended, the remaining supply of ready processes in systems where all suspended processes remains resident in memory may become reduced to a level that impairs functioning of schedulers.

• The medium term scheduler is in charge of handling the swapped out processes.

• It has little to do while a process is remained as suspended.

Short Term Scheduler

• The short term scheduler allocates the processor among the pool of ready processes resident in the memory.

• Its main objective is to maximize system performance in accordance with the chosen set of criteria.

• Some of the events introduced thus for that cause rescheduling by virtue of their ability to change the global system state are:

• Clock ticks

• Interrupt and I/O completions

• Most operational OS calls

• Sending and receiving of signals

• Activation of interactive programs.

• Whenever one of these events occurs ,the OS involves the short term scheduler.

Scheduling Criteria :

- **CPU Utilization:**

  Keep the CPU as busy as possible. It range from 0 to 100%. In practice, it range from 40 to 90%.

- **Throughput:**

  Throughput is the rate at which processes are completed per unit of time.

- **Turnaround time:**

  This is the how long a process takes to execute a process. It is calculated as the time gap between the submission of a process and its completion.

- **Waiting time:**

  Waiting time is the sum of the time periods spent in waiting in the ready queue.

- **Response time:**

  Response time is the time it takes to start responding from submission time. It is calculated as the amount of time it takes from when a request was submitted until the first response is produced.

Non-preemptive Scheduling :

In non-preemptive mode, once if a process enters into running state, it continues to execute until it terminates or blocks itself to wait for Input/Output or by requesting some operating system service.

Preemptive Scheduling :

In preemptive mode, currently running process may be interrupted and moved to the ready State by the operating system.
When a new process arrives or when an interrupt occurs, preemptive policies may incur greater overhead than non-preemptive version but preemptive version may provide better service.

It is desirable to maximize CPU utilization and throughput, and to minimize turnaround time, waiting time and response time.

Types of scheduling Algorithms

• In general, scheduling disciplines may be pre-emptive or non-pre-emptive .

• In batch, non-pre-emptive implies that once scheduled, a selected job turns to completion.

There are different types of scheduling algorithms such as:

- FCFS(First Come First Serve)
- SJF(Short Job First)
- Priority scheduling
- Round Robin Scheduling algorithm

First Come First Serve Algorithm

• FCFS is working on the simplest scheduling discipline.

• The workload is simply processed in an order of their arrival, with no pre-emption.

• FCFS scheduling may result into poor performance.

• Since there is no discrimination on the basis of required services, short jobs may considerable in turn around delay and waiting time.

Advantages

- Better for long processes
- Simple method (i.e., minimum overhead on processor)
- No starvation

Disadvantages

- Convoy effect occurs. Even very small process should wait for its turn to come to utilize the CPU. Short process behind long process results in lower CPU utilization.
- Throughput is not emphasized.

First Come, First Served

Process	Burst Time
P1	24
P2	3
P3	3

- Suppose that the processes arrive in the order: *P1* , *P2* , *P3*
- The Gantt Chart for the schedule is:

- Waiting time for *P1* = 0; *P2* = 24; *P3* = 27
- Average waiting time: (0 + 24 + 27)/3 = 17

Note : solve complete e.g. as we studied in practical(above is just sample e.g.). you can take any e.g.

Shortest Job First Algorithm :

- his is also known as **shortest job first**, or SJF
- This is a non-preemptive, pre-emptive scheduling algorithm.
- Best approach to minimize waiting time.
- Easy to implement in Batch systems where required CPU time is known in advance.
- Impossible to implement in interactive systems where required CPU time is not known.
- The processer should know in advance how much time process will take.

Advantages

- It gives superior turnaround time performance to shortest process next because a short job is given immediate preference to a running longer job.
- Throughput is high.

Disadvantages

- Elapsed time (i.e., execution-completed-time) must be recorded, it results an additional overhead on the processor.
- Starvation may be possible for the longer processes.

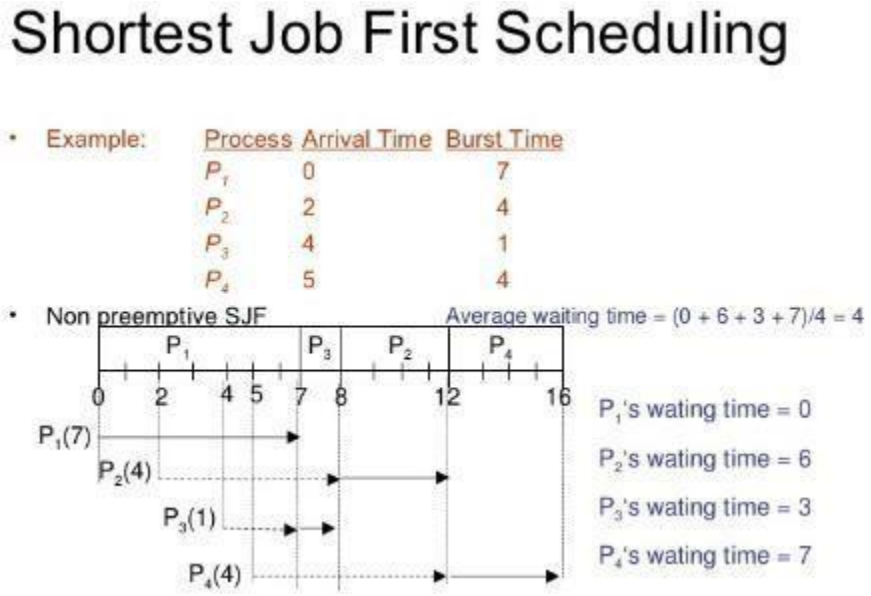
This algorithm is divided into two types:

• Pre-emptive SJF

• Non-pre-emptive SJF

• **Pre-emptive SJF Algorithm:**

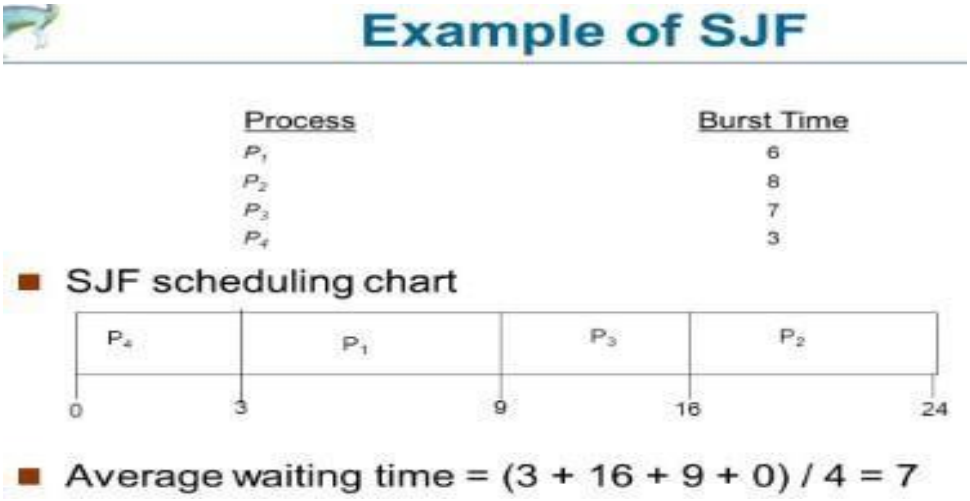
In this type of SJF, the shortest job is executed 1st. the job having least arrival time is taken first for execution. It is executed till the next job arrival is reached.



Note : solve complete e.g. as we studied in practical(above is just sample e.g.). you can take any e.g.

Non-pre-emptive SJF Algorithm:

In this algorithm, job having less burst time is selected 1st for execution. It is executed for its total burst time and then the next job having least burst time is selected.



Note : solve complete e.g. as we studied in practical(above is just sample e.g.). you can take any e.g.

Round Robin Scheduling :

- Round Robin is the preemptive process scheduling algorithm.
- Each process is provided a fix time to execute, it is called a **quantum**.
- Once a process is executed for a given time period, it is preempted and other process executes for a given time period.
- Context switching is used to save states of preempted processe

Advantages

- Round-robin is effective in a general-purpose, times-sharing system or transaction-processing system.
- Fair treatment for all the processes.
- Overhead on processor is low.
- Overhead on processor is low.
- Good response time for short processes.

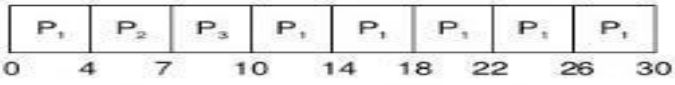
Disadvantages

- Care must be taken in choosing quantum value.
- Processing overhead is there in handling clock interrupt.
- Throughput is low if time quantum is too small.

Round Robin

Process	Burst Time
P1	24
P2	3
P3	3

- Quantum time = 4 milliseconds
- The Gantt chart is:



- Average waiting time = {[0+(10-4)]+4+7}/3 = 5.6

Note : solve complete e.g. as we studied in practical(above is just sample e.g.). you can take any e.g.

Priority Scheduling :

- Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems.
- Each process is assigned a priority. Process with highest priority is to be executed first and so on.
- Processes with same priority are executed on first come first served basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Advantage

- Good response for the highest priority processes.

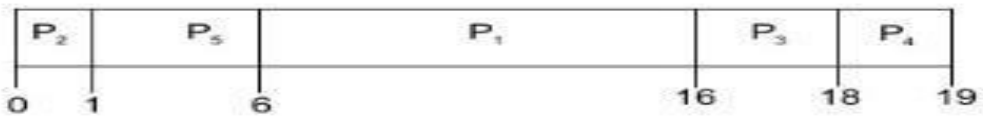
Disadvantage

- Starvation may be possible for the lowest priority processes.

Priority

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

- Gantt Chart



- Average waiting time = (6 + 0 + 16 + 18 + 1)/5 = 8.2

Note : solve complete e.g. as we studied in practical(above is just sample e.g.). you can take any e.g.

8. Algorithms(procedure) :

FCFS :

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Set the waiting of the first process as '0' and its burst time as its turn around time

Step 5: for each process in the Ready Q calculate

(a) Waiting time for process(n)= waiting time of process (n-1) + Burst time of process(n-1)

(b) Turn around time for Process(n)= waiting time of Process(n)+ Burst time for process(n)

Step 6: Calculate

(a) Average waiting time = Total waiting Time / Number of process

(b) Average Turnaround time = Total Turnaround Time / Number of process

Step 7: Stop the process

SJF :

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Start the Ready Q according the shortest Burst time by sorting according to lowest to highest burst time.

Step 5: Set the waiting time of the first process as '0' and its turnaround time as its burst time.

Step 6: For each process in the ready queue, calculate

(c) Waiting time for process(n)= waiting time of process (n-1) + Burst time of process(n-1)

(d) Turn around time for Process(n)= waiting time of Process(n)+ Burst time for process(n)

Step 6: Calculate

(c) Average waiting time = Total waiting Time / Number of process

(d) Average Turnaround time = Total Turnaround Time / Number of process

Step 7: Stop the process

RR :

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue and time quantum (or) time slice

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Calculate the no. of time slices for each process where

No. of time slice for process(n) = burst time process(n)/time slice

Step 5: If the burst time is less than the time slice then the no. of time slices =1.

Step 6: Consider the ready queue is a circular Q, calculate

(a) Waiting time for process(n) = waiting time of process(n-1)+ burst time of process(n-1) + the time difference in getting the CPU from process(n-1)

(b) Turn around time for process(n) = waiting time of process(n) + burst time of process(n)+ the time difference in getting CPU from process(n).

Step 7: Calculate

(e) Average waiting time = Total waiting Time / Number of process

(f) Average Turnaround time = Total Turnaround Time / Number of process

Step 8: Stop the process.

Priority Scheduling :

Algorithms :

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time, priority

Step 4: Start the Ready Q according the priority by sorting according to lowest to highest burst time and process.

Step 5: Set the waiting time of the first process as '0' and its turnaround time as its burst time.

``

Step 6: For each process in the ready queue, calculate

(e) Waiting time for process(n)= waiting time of process (n-1) + Burst time of process(n-1)

(f) Turn around time for Process(n)= waiting time of Process(n)+ Burst time for process(n)

Step 6: Calculate

(g) Average waiting time = Total waiting Time / Number of process

(h) Average Turnaround time = Total Turnaround Time / Number of process

Step 7: Stop the process

Note: you can write algorithm & procedure as per your program/concepts

9. Flowchart :

Note: you should draw flowchart as per algorithm/procedure as above

10. Conclusion:

Hence we have studied and implemented that-

- CPU scheduling concepts like context switching, types of schedulers, different timing parameter like waiting time, turnaround time, burst time, etc.
- Different CPU scheduling algorithms like FIFO, SJF, Priority and Round Robin Etc.

GROUP – B-7

EXPERIMENT NO: 05

1. **Title:**

   Write a program to simulate Page replacement algorithm..

2. **Objectives :**

   - To understand Page replacement policies
   - To understand paging concept
   - To understand Concept of page fault, page hit, miss, hit ratio etc

3. **Problem Statement :**

   Write a java program to implement Page Replacement Algorithm FIFO, LRU and OPT.

4. **Outcomes:**

After completion of this assignment students will be able to:

   - Knowledge of Page Replacement Policies in OS
   - Implemented LRU & OPT Page replacement Policies
   - Understood concept of paging.

5. **Software Requirements:**

   Latest jdk., Eclipse

6. **Hardware Requirement:**

   - M/C Lenovo Think center M700 Ci3,6100,6th Gen. H81, 4GB RAM ,500GB HDD

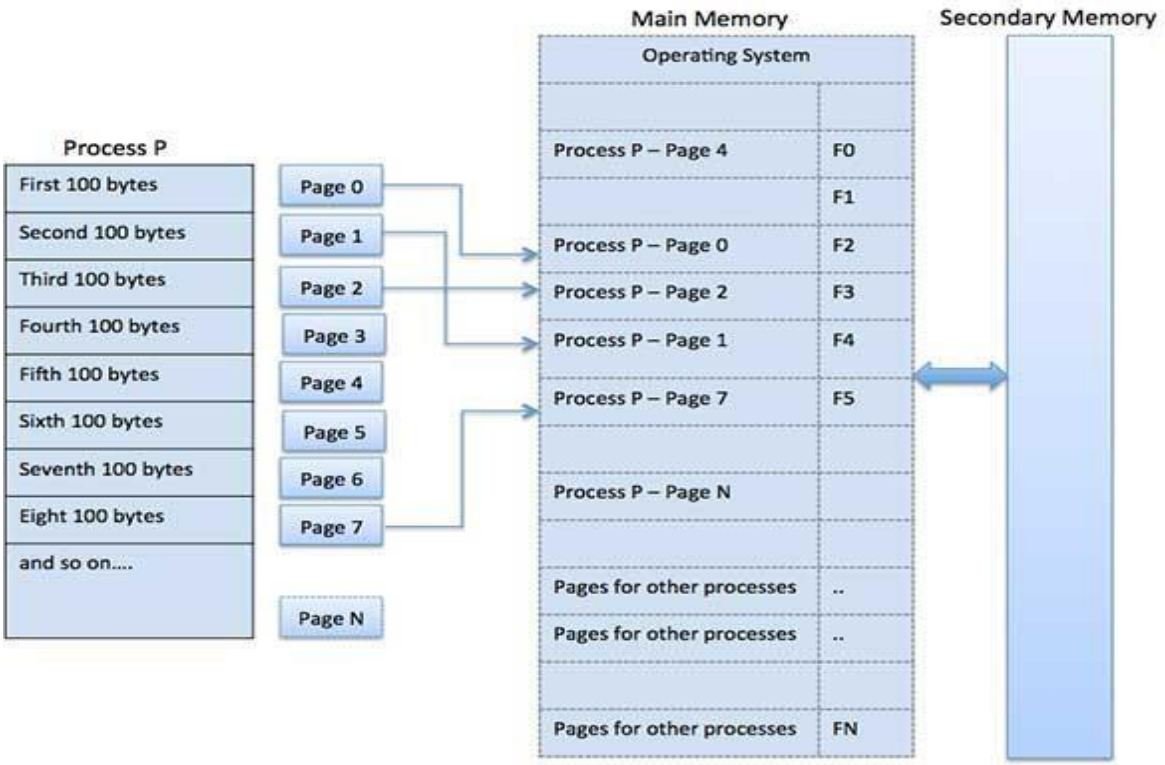
7. **Theory Concepts:**

Paging :

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard that's set up to emulate the computer's RAM. Paging technique plays an important role in implementing virtual memory.

Paging is a memory management technique in which process address space is broken into blocks of the same size called **pages** (size is power of 2, between 512 bytes and 8192 bytes). The size of the process is measured in the number of pages.

Similarly, main memory is divided into small fixed-sized blocks of (physical) memory called **frames** and the size of a frame is kept the same as that of a page to have optimum utilization of the main memory and to avoid external fragmentation.



Address Translation

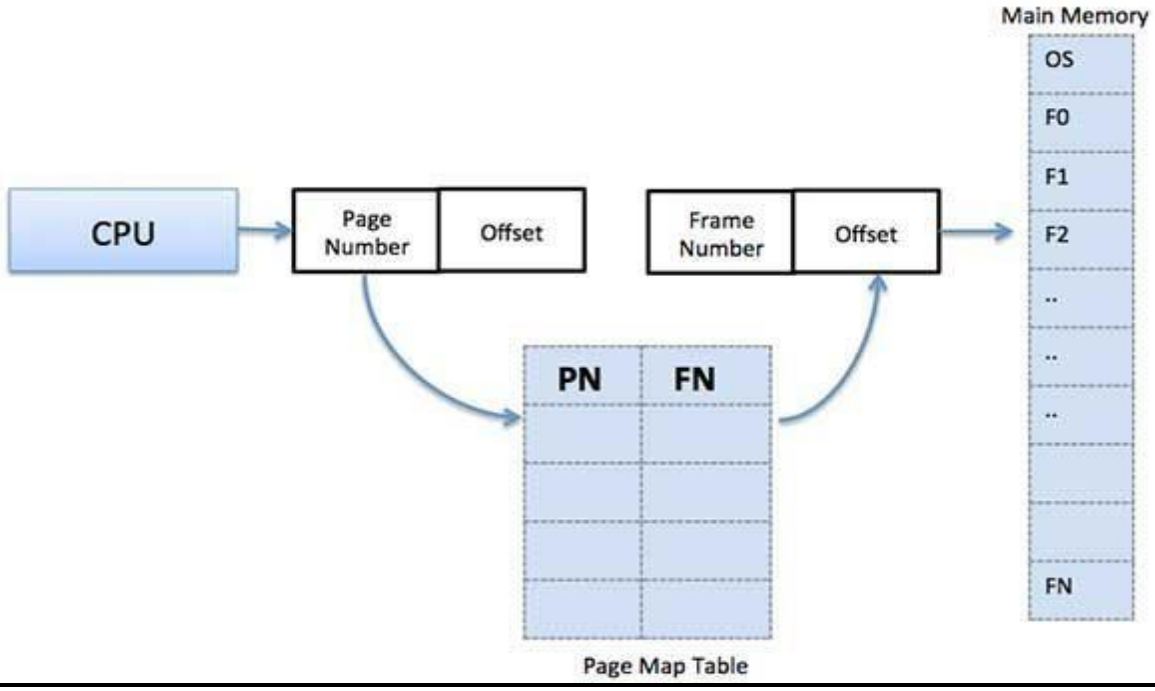
Page address is called **logical address** and represented by **page number**and the **offset**.

Logical Address = Page number + page offset

Frame address is called **physical address** and represented by a **frame number** and the **offset**.

Physical Address = Frame number + page offset

A data structure called **page map table** is used to keep track of the relation between a page of a process to a frame in physical memory.



When the system allocates a frame to any page, it translates this logical address into a physical address and create entry into the page table to be used throughout execution of the program.

When a process is to be executed, its corresponding pages are loaded into any available memory frames. Suppose you have a program of 8Kb but your memory can accommodate only 5Kb at a given point in time, then the paging concept will come into picture. When a computer runs out of RAM, the operating system (OS) will move idle or unwanted pages of memory to secondary memory to free up RAM for other processes and brings them back when needed by the program.

This process continues during the whole execution of the program where the OS keeps removing idle pages from the main memory and write them onto the secondary memory and bring them back when required by the program.

Advantages and Disadvantages of Paging

Here is a list of advantages and disadvantages of paging −

- Paging reduces external fragmentation, but still suffer from internal fragmentation.
- Paging is simple to implement and assumed as an efficient memory management technique.
- Due to equal size of the pages and frames, swapping becomes very easy.
- Page table requires extra memory space, so may not be good for a system having small RAM.

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called **virtual memory** and it is a section of a hard disk that's set up to emulate the computer's RAM.

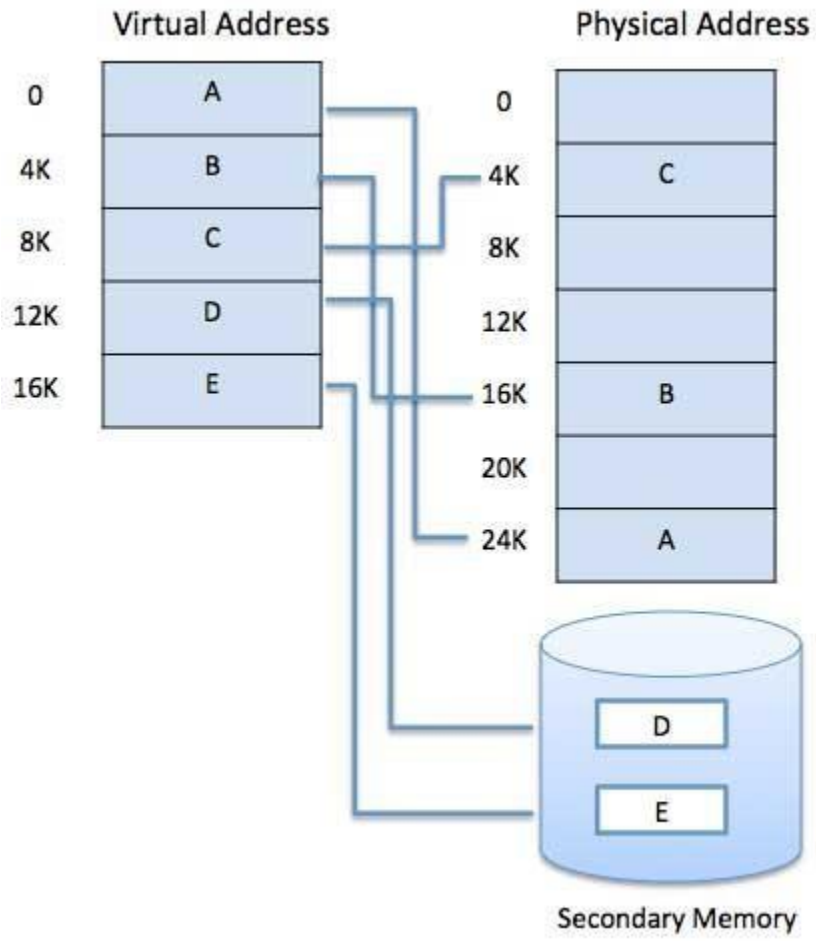
The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes. First, it allows us to extend the use of physical memory by using disk. Second, it allows us to have memory protection, because each virtual address is translated to a physical address.

Following are the situations, when entire program is not required to be loaded fully in main memory.

- User written error handling routines are used only when an error occurred in the data or computation.
- Certain options and features of a program may be used rarely.
- Many tables are assigned a fixed amount of address space even though only a small amount of the table is actually used.
- The ability to execute a program that is only partially in memory would counter many benefits.
- Less number of I/O would be needed to load or swap each user program into memory.
- A program would no longer be constrained by the amount of physical memory that is available.

- Each user program could take less physical memory, more programs could be run the same time, with a corresponding increase in CPU utilization and throughput.

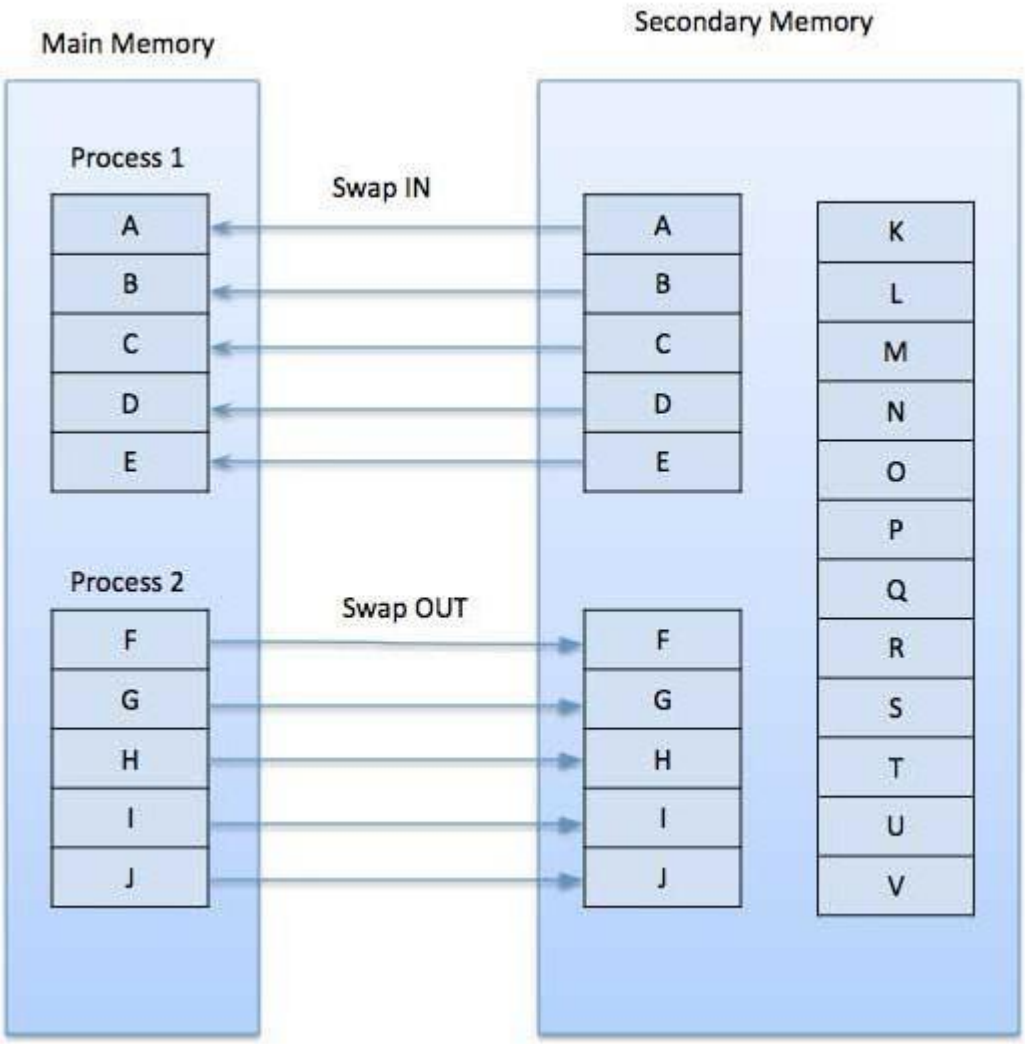
Modern microprocessors intended for general-purpose use, a memory management unit, or MMU, is built into the hardware. The MMU's job is to translate virtual addresses into physical addresses. A basic example is given below −



Virtual memory is commonly implemented by demand paging. It can also be implemented in a segmentation system. Demand segmentation can also be used to provide virtual memory.

Demand Paging

A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand, not in advance. When a context switch occurs, the operating system does not copy any of the old program’s pages out to the disk or any of the new program’s pages into the main memory Instead, it just begins executing the new program after loading the first page and fetches that program’s pages as they are referenced.



While executing a program, if the program references a page which is not available in the main memory because it was swapped out a little ago, the processor treats this invalid memory reference as a **page fault** and transfers control from the program to the operating system to demand the page back into the memory.

Advantages

Following are the advantages of Demand Paging −

- Large virtual memory.
- More efficient use of memory.
- There is no limit on degree of multiprogramming.

Disadvantages

- Number of tables and the amount of processor overhead for handling page interrupts are greater than in the case of the simple paged management techniques.

Page Replacement Algorithm:

Page replacement algorithms are the techniques using which an Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated. Paging happens whenever a page fault occurs and a free page cannot be used for allocation purpose accounting to reason that pages are not available or the number of free pages is lower than required pages.

When the page that was selected for replacement and was paged out, is referenced again, it has to read in from disk, and this requires for I/O completion. This process determines the quality of the page replacement algorithm: the lesser the time waiting for page-ins, the better is the algorithm.

A page replacement algorithm looks at the limited information about accessing the pages provided by hardware, and tries to select which pages should be replaced to minimize the total number of page misses, while balancing it with the costs of primary storage and processor time of the algorithm itself. There are many different page replacement algorithms. We evaluate an algorithm by running it on a particular string of memory reference and computing the number of page faults,

Page fault :

A **page fault** (sometimes called #PF, PF or hard **fault**) is a type of exception raised by computer hardware when a running program accesses a memory **page**that is not currently mapped by the memory management unit (MMU) into the virtual address space of a process.

Page hit :

A **hit** is a request to a web server for a file, like a web **page**, image, JavaScript, or Cascading Style Sheet. When a web **page** is downloaded from a server the number of "**hits**" or "**page hits**" is equal to the number of files requested.

Page frame :

The **page frame** is the storage unit (typically 4KB in size) whereas the **page** is the contents that you would store in the storage unit ie the **page frame**. For eg) the RAM is divided into fixed size blocks called **page frames** which is typically 4KB in size, and each **page frame** can store 4KB of data ie the **page**.

Page table :

A **page table** is the data structure used by a virtual memory system in a computer operating system to store the mapping between virtual addresses and physical addresses.

Reference String :

The string of memory references is called reference string. Reference strings are generated artificially or by tracing a given system and recording the address of each memory reference. The latter choice produces a large number of data, where we note two things.

- For a given page size, we need to consider only the page number, not the entire address.

- If we have a reference to a page **p**, then any immediately following references to page **p** will never cause a page fault. Page p will be in memory after the first reference; the immediately following references will not fault.

- For example, consider the following sequence of addresses − 123,215,600,1234,76,96
- If page size is 100, then the reference string is 1,2,6,12,0,0

First In First Out (FIFO) algorithm:

- Oldest page in main memory is the one which will be selected for replacement.
- Easy to implement, keep a list, replace pages from the tail and add new pages at the head.

Advantages

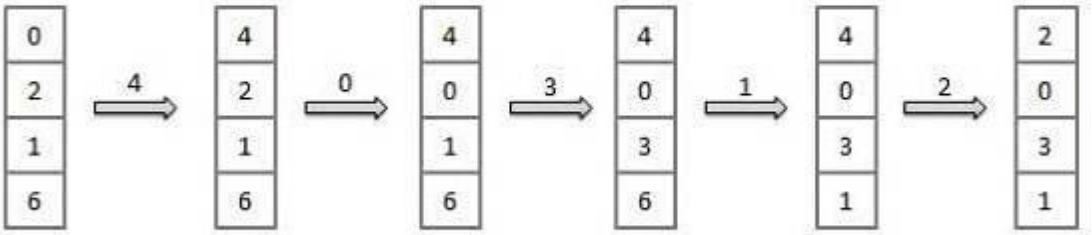
1. Simple to understand and implement
2. Does not cause more overhead

Disadvantages

1. Poor performance
2. Doesn't use the frequency of the last used time and just simply replaces the oldest page.
3. Suffers from Belady's anomaly.

Reference String : 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses : x x x x x x x x x



Fault Rate = 9 / 12 = 0.75

Page reference stream:

1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
1	1	1	1	1	2	2	3	5	1	6	6	2	5	5	3	3	1	6	2
	2	2	2	2	3	3	5	1	6	2	2	5	3	3	1	1	6	2	4
		3	3	3	5	5	1	6	2	5	5	3	1	1	6	6	2	4	3
*	*	*			*		*	*	*	*		*	*		*		*	*	*

FIFO
Total 14 page faults

Note : you can take other example also. This just for reference. (you must calculate page fault, page hit and hit ratio)

Least Recently Used (LRU) algorithm:

- Page which has not been used for the longest time in main memory is the one which will be selected for replacement.
- Easy to implement, keep a list, replace pages by looking back into time.

Advantages

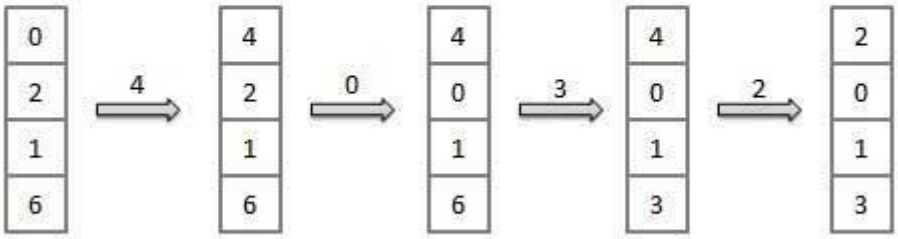
1. It is open for full analysis
2. Doesn't suffer from Belady's anomaly
3. Often more efficient than other algorithms

Disadvantages

1. It requires additional data structures to be implemented
2. More complex
3. High hardware assistance is required

Reference String : 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses : x x x x x x x x



Fault Rate = 8 / 12 = 0.67

Page reference stream:

1 2 3 2 1 5 2 1 6 2 5 6 3 1 3 6 1 2 4 3

1	1	1	1	3	2	1	5	2	1	6	2	5	6	6	1	3	6	1	2
	2	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4
		3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
*	*	*			*			*		*		*	*				*	*	*

LRU

Total 11 page faults

Note : you can take other example also. This just for reference. (you must calculate page fault, page hit and hit ratio)

Optimal Page algorithm:

- An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms. An optimal page-replacement algorithm exists, and has been called OPT or MIN.
- Replace the page that will not be used for the longest period of time. Use the time when a page is to be used.

Advantages

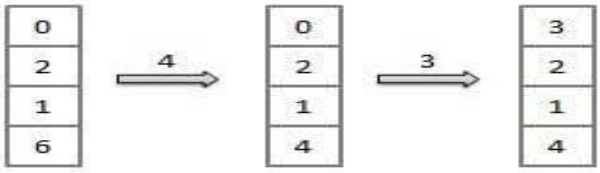
1. Excellent efficiency
2. Less complexity
3. Easy to use and understand
4. Simple data structures can be used to implement
5. Used as the benchmark for other algorithms

Disadvantages

1. More time consuming
2. Difficult for error handling
3. Need future awareness of the programs, which is not possible every time

Reference String : 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses : x x x x x x



Fault Rate = 6 / 12 = 0.50

Page reference stream:

1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
1	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	2	2	2
	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	4	4
		3	3	3	5	5	5	5	5	5	5	3	3	3	3	3	3	3	3
*	*	*			*			*				*	*				*	*	

Optimal

Total 9 page faults

Note : you can take other example also. This just for reference. (you must calculate page fault, page hit and hit ratio)

Page Buffering algorithm

- To get a process start quickly, keep a pool of free frames.
- On page fault, select a page to be replaced.
- Write the new page in the frame of free pool, mark the page table and restart the process.
- Now write the dirty page out of disk and place the frame holding replaced page in free pool.

Least frequently Used(LFU) algorithm

- The page with the smallest count is the one which will be selected for replacement.
- This algorithm suffers from the situation in which a page is used heavily during the initial phase of a process, but then is never used again.

Most frequently Used(MFU) algorithm

- This algorithm is based on the argument that the page with the smallest count was probably just brought in and has yet to be used.

8. Conclusion :

Thus , We have implemented page replacement policies FIFO, LRU and OPT.

Part II: Elective I

Software Project Management
Lab Practice I (310248)
Lab Manual

CONTENTS

Group A

1. Create Project Plan
   - Specify project name and start (or finish) date.
   - Identify and define project tasks.
   - Define duration for each project task.
   - Define milestones in the plan
   - Define dependency between tasks
   - Define project calendar.
   - Define project resources and specify resource type
   - Assign resources against each task and baseline the project plan

2. Execute and Monitor Project Plan
   - Update % Complete with current task status.
   - Review the status of each task.
   - Compare Planned vs Actual Status
   - Review the status of Critical Path
   - Review resources assignation status

3. Generate Dashboard and Reports
   - Dashboard
     - Project Overview
     - Cost Overview
     - Upcoming Tasks
   - Resource Reports
     - Over-allocated Resources
     - Resource Overview
   - Cost Reports
     - Earned Value Report
     - Resource Cost Overview
     - Task Cost Overview
   - Progress Reports
     - Critical Tasks
     - Milestone Report
     - Slipping Tasks

Schedule Estimation using Gantt Chart.

Aim: Schedule Estimation using Gantt Chart

Theory:

Introduction

Gantt chart is a type of a bar chart that is used for illustrating project schedules. Gantt charts can be used in any projects that involve effort, resources, milestones and deliveries.

At present, Gantt charts have become the popular choice of project managers in every field. Gantt charts allow project managers to track the progress of the entire project. Through Gantt charts, the project manager can keep a track of the individual tasks as well as of the overall project progression.

In addition to tracking the progression of the tasks, Gantt charts can also be used for tracking the utilization of the resources in the project. These resources can be human resources as well as materials used.

Gantt chart was invented by a mechanical engineer named Henry Gantt in 1910. Since the invention, Gantt chart has come a long way. By today, it takes different forms from simple paper based charts to sophisticated software packages.

The underlying concept of a Gantt chart is to map out which tasks can be done in parallel and which need to be done sequentially. If we combine this with the project resources we can explore the trade-off between the scope (doing more or less work), cost (using more or less resources) and the time scales for the project. By adding more resources or reducing the scope the project manager can see the effect on the end date.

A Gantt chart displays information visually as a type of bar chart in a clear and easy-to- understand way and is used for the following activities:

- Establish the initial project schedule
- Allocate resources
- Monitor and report progress
- Control and communicate the schedule
- Display milestones
- Identify and report problems

To create a chart you need to know all of the individual tasks required to complete the project, an estimate of how long each task will take and which tasks are dependent on others. The very process of pulling this information together helps a project manager focus on the essential parts of the project and begin to establish a realistic timeframe for completion.

The Use

As we have already discussed, Gantt charts are used for project management purposes. In order to use Gantt charts in a project, there are a few initial requirements fulfilled by the project.

First of all, the project should have a sufficiently detailed Work Breakdown Structure (WBS). Secondly, the project should have identified its milestones and deliveries.

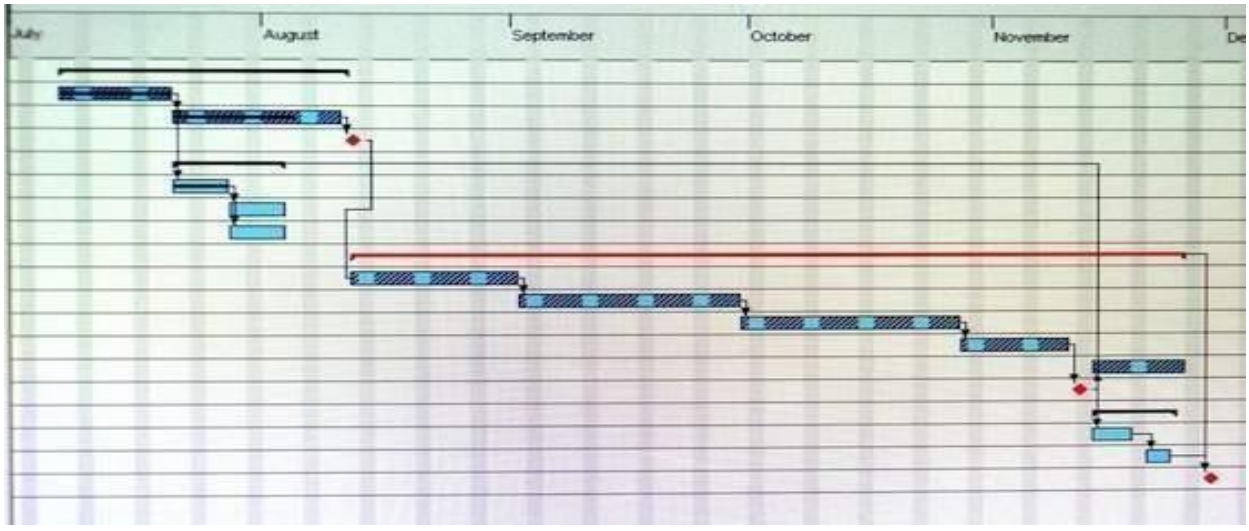
In some instances, project managers try to define the work break down structure while creating Gantt chart. This is one of the frequently practised errors in using Gantt charts. Gantt charts are not designed to assist WBS process; rather Gantt charts are for task progress tracking.

Gantt charts can be successfully used in projects of any scale. When using Gantt charts for large projects, there can be an increased complexity when tracking the tasks.

This problem of complexity can be successfully overcome by using computer software packages designed for offering Gantt chart functionalities.

Tools Available

There are dozens of Gantt chart tools that can be used for successful project tracking. These tools usually vary by the feature offered.



The simplest kind of Gantt chart can be created using a software tool such as Microsoft Excel. For that matter, any spreadsheet tool can be used to design a Gantt chart template.
If the project is small scale and does not involve many parallel tasks, a spreadsheet based Gantt chart can be the most effective type.

Microsoft Project is one of the key Gantt chart tools used today. Especially for software development projects, MS Project based Gantt charts are essential to track the hundreds of parallel tasks involved in the software development life cycle.
There are many other Gantt chart tools available for free and for price. The features offered by these tools range from the same features offered by Excel based Gantt charts to MS Project Gantt charts. These tools come with different price tags and feature levels, so one can select the suitable Gantt chart tool for the purpose in hand.

Creating Your Own
Sometimes, one may decide to create their own Gantt chart tool without buying an existing one. If this is the case, first of all, one should search the Internet for free Gantt chart templates.

This way, one may actually find the exact Gantt chart template (probably in Excel) required for the purpose. In case, if no match is found, then it is sensible to create one's own.
Excel is the most popular tool for creating custom Gantt charts. Of course, one can create a Gantt chart from scratch in Excel, but it is always advisable to use a Project Management add-on in Excel to create Gantt charts. These project management add-ons are published by Microsoft and other third-party companies.

Why Use Gantt Charts?

When you set up a Gantt chart, you need to think through all of the tasks involved in your project. As part of this process, you'll work out who will be responsible for each task, how long each task will take, and what problems your team may encounter.

This detailed thinking helps you ensure that the schedule is workable, that the right people are assigned to each task, and that you have workarounds for potential problems before you start.

They also help you work out practical aspects of a project, such as the minimum time it will take to deliver, and which tasks need to be completed before others can start. Plus, you can use them to identify the critical path – the sequence of tasks that must individually be completed on time if the whole project is to deliver on time.

Finally, you can use them to keep your team and your sponsors informed of progress. Simply update the chart to show schedule changes and their implications, or use it to communicate that key tasks have been completed.

Advantages & Disadvantages:

The ability to grasp the overall status of a project and its tasks at once is the key advantage in using a Gantt chart tool. Therefore, upper management or the sponsors of the project can make informed decisions just by looking at the Gantt chart tool.

The software-based Gantt charts are able to show the task dependencies in a project schedule. This helps to identify and maintain the critical path of a project schedule.

Gantt chart tools can be used as the single entity for managing small projects. For small projects, no other documentation may be required; but for large projects, the Gantt chart tool should be supported by other means of documentation.

For large projects, the information displayed in Gantt charts may not be sufficient for decision making. Although Gantt charts accurately represent the cost, time and scope aspects of a project, it does not elaborate on the project size or size of the work elements. Therefore, the magnitude of constraints and issues can be easily misunderstood.

Conclusion :

Gantt chart tools make project manager's life easy. Therefore, Gantt chart tools are important for successful project execution. Identifying the level of detail required in the project schedule is the key when selecting a suitable Gantt chart tool for the project.

Group A

EXPERIMENT No: 1

Problem Statement :

Create Project Plan

- Specify project name and start (or finish) date.
- Identify and define project tasks.
- Define duration for each project task.
- Define milestones in the plan
- Define dependency between tasks
- Define project calendar.
- Define project resources and specify a resource type
- Assign resources against each task and baseline the project plan

Objectives :

- To learn various techniques , tools , applications in software project management .
- To understand effective communication , collaboration , and productive guidelines .
- Meet the exclusive needs and requirements of client .
- Achieve project goals within the estimated time with high quality .
- Development and Implementation of Procedure .

Theory :

Project Plan :

A **project pla**n is a series of formal documents that define the execution and control stages of a project. The plan includes considerations for risk management, resource management and communications, while also addressing scope, cost and schedule baselines. Project planning software is used by project managers to ensure that their plans are thorough and robust.

Elements of Project planning :

How to create a project calender ?

Step 1 - Review Scope Baseline

Gather the team and review the approved scope baseline, which consists of three components:

1) the Scope Statement

2) the Work Breakdown Structure (WBS) and the WBS Dictionary.

3) The project team member should confirm that the scope baseline addresses 100% of the project scope.

Step 2 - Create Activities

Using a technique called Decomposition, the project team breaks down each WBS work package into activities. Just like when creating the WBS work packages, the team needs to set rules for creating schedule activities. The final schedule needs to be the one that is effective and efficient. Too many activities can be as bad as too few. It is also important to identify deadlines and milestones while decomposing the project.

Step 3 - Sequence Activities

Every activity is related to one or more other activities. Every activity, except the first and last, has a relationship with a predecessor and a successor. Sequencing activities means placing the activities in the right order using the right relationships. There are four types of relationships:

- **1. Finish to Start –** Cannot start the successor activity until its predecessor Is finished.

- **2. Start to Start –** Cannot start the successor activity until its predecessor has started.
- **3. Start to Finish –** Cannot finish the successor activity until its predecessor had started.
- **4. Finish to Finish –** Cannot finish the successor activity until its predecessor has finished.

Relationships 1 and 2 are the most commonly used. Finish to Start is a sequential relationship and Start to Start is typically a parallel or over-lapping relationship.

Step 4 - Estimate Resources :

Before the durations can be estimated, resources must be identified and estimated. Resources include labor, material and equipment. There are several estimating techniques used including Analogous, Parametric, Three-Point and Bottom Up. Skills, competencies and technology are key factors to consider in the basis of the estimate. After estimating the resources, they are loaded in the schedule against the respective activities. A resource calendar is also created to show when resources are needed and available.

Step 5 - Estimate Durations :

Duration is the time between the start and end of an activity. Review the resources, relationships and sequencing, then estimate the duration for each activity. The same estimating techniques used for estimating resources can be used to estimate durations, but make sure you identify constraints. Which are limitations or restrictions on an activity.

Step 6 - Develop Schedule

Create the Gantt chart by loading all information into a project management software tool. Review the schedule and ensure that all schedule risks have been addressed. Check that response plans and schedule contingencies have been included. A typical way to address schedule contingencies is to add Buffers at the activity level, the project level or both. A Buffer is an activity with no resources or scope to provide additional time and reduce schedule risks. Resource optimization techniques, such as resource smoothing or leveling are used to create realistic schedules. Review and approve the schedule. The approved Gantt chart schedule becomes the schedule baseline

Input :

1) Project or program charter .
2) Baseline for Scope , Schedule , and Cost .
3) Requirements
4) Agreements
5) Management plans for Scope , Schedule , Cost , Quality , Human Resources , Communication , Risk and Procurement .

Output :

1) Deliverables.
2) Work Performance Data.
3) Issue Log.
4) Change Requests.
5) Project Management Plan Update.
6) Project Document Update
7) Organization process Assets Updates

Testcases :

1) Completeness
2) Clarity
3) Sufficient Data
4) Reference to dependent cases

Software Requirements :

1) MS Project
2) Gantt Project
3) Primavera

Hardware Requirements : N/A

Output of Primavera

Frequently Asked Questions :

1) What is PERT in Project Management ?

➔ A PERT Chart is a Project Management Tool that provides a graphical representation of a project timeline

➔ The Program Evaluation Review Technique (PERT) break down the individual task of project for analysis.

2) What is Resource Allocation in Project Management ?

➔ Resource Allocation is the process of assigning and scheduling available resource in the most effective and economical way possible .

➔ It is the Management and delegation of resources throughout the project to ensure that it runs as smoothly and successfully as possible .

3) What is the Gantt Chart in Project Management ?

➔ A Gantt Chart is the Project Management Tool assisting in the planning and scheduling of projects of all sizes , although , they are particularly useful for simplifying complex project .

➔ As its in a bar chart format it is possible to check on progress with a quick glance

4) What are the main factors to consider when deciding whether to build or purchase software ?

➔ When determining whether to purchase software or begin building software, you must consider many factors about your organization, your available resources, and the project scope the software is intended to resolve. These include the number of internal and external software users, how quickly the problem needs to be resolved, who you can assign to deploy a solution, how many processes will be affected by a new software, and the cost of a solution.

5) What is WBS ?

➔ A work breakdown structure is a project management tool used to define and manage a project's deliverables. The WBS is a hierarchical structure that breaks down complex activities into more manageable parts, allowing users to see the individual deliverables that need to be completed in order to reach a project's overarching goal.

➔ While most project management tools focus on planned actions, a WBS focuses on planned outcomes. A carefully organized WBS can help a project manager more effectively oversee the completion of otherwise complicated tasks within a project. A WBS with measurable, clearly defined tasks can also help project management assign accurate costs and deadlines to a project, simplifying project planning and monitoring.

EXPERIMENT No: 2

Problem Statement :

Execute and Monitor Project Plan

- Update % Complete with current task status.
- Review the status of each task.
- Compare Planned vs Actual Status
- Review the status of the Critical Path
- Review resources assignation status

Objectives :

To learn various techniques, tools, applications in software project management .
To understand efficient methods for planning the projects.
To monitor and execute projects with a plan.

Theory :

Software Project Management is dedicated to the planning, scheduling, resource allocation, execution tracking, and delivery of software and projects.

The project management processes consist of initiating the project. Here the project manager writes the business case and project charter. The project manager also carries out the planning processes that completes the work breakdown structure and performs project schedule cost estimation and so on.

Then the project manager carries out execution processes, which performs necessary action to complete work as outlined in the plan , but then during the execution process there may be deviations from the plan .

The project manager needs to carry out the monitoring and control processes where the project manager checks whether project is proceeding as plan or there are deviations . The project manager takes corrective actions to the progress of project with the plan and finally , the project manager needs to carry out closing processes , where closing documents are created and the customer finally gives the formal acceptance of project ,

Any project, whether big or small, has the potential to be very complex. It's much easier to break down all the necessary inclusions for a project plan by viewing your project in terms of phases. The Project Management Institute, within the Project Management Book of Knowledge (PMBOK), have identified the following 5 phases of a project:

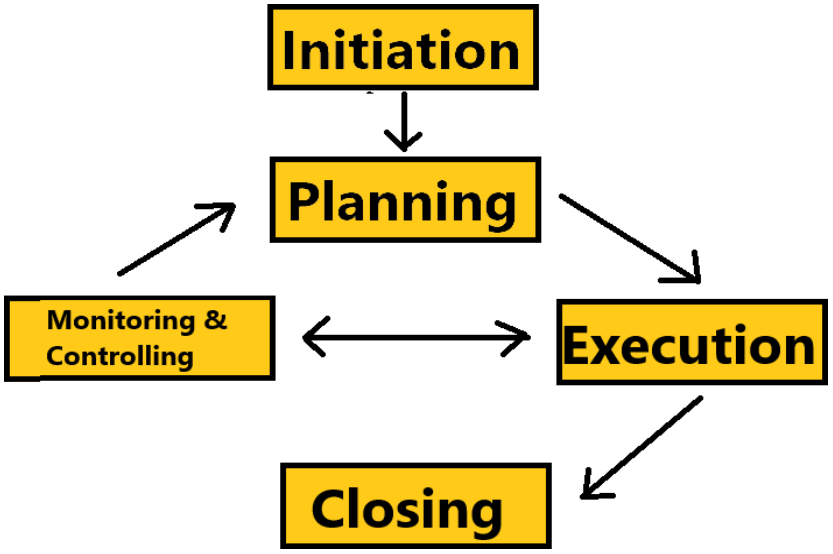
- **Initiation**: The start of a project, in which goals and objectives are defined through a business case and the practicality of the project is determined by a feasibility study.

- **Planning**: During the project planning phase, the scope of the project is defined by a work breakdown structure (WBS) and the project methodology to manage the project is decided on. Costs, quality and resources are estimated, and a project schedule with milestones and task dependencies is identified. The main deliverable of this phase is your project plan.

- **Execution**: The project deliverables are completed during this phase. Usually, this phase begins with a kick-off meeting and is followed by regular team meetings and status reports while the project is being worked on.

- **Monitoring & Controlling**: This phase is performed in tandem with the project execution phase. Progress and performance metrics are measured to keep progress on the project aligned with the project plan.

- **Closure**: The project is completed when the stakeholder receives the final deliverable. Resources are released, contracts are signed off on and, ideally, there will be an evaluation of the successes and failures.



The project manager has three main objectives during the execution phase :

1) Manages People
2) Manages Process
3) Manages communication

The benefits of a well-executed project are threefold :

- The project can be completetd on time and budget
- Team moral can be maintained
- Stakeholders are satisfied with overall project progress

The 6 phases of project monitoring are :

- Identify goals of a project
- Define the indicators

- Define data collection methods and training
- Identify roles and responsibilities during monitoring
- Create an analysis plan and report templates
- Plan data disclosure

Input :

1) Project Management Plan.
2) Project Document.
3) Work Breakdown Structure (WBS).

Output :

- Project deliverables are tangible outputs of project . They need to be reviewed and meet acceptance by clients.
- Change Requests : When client expectations change or there's a disconnect between team members.
- The execution stage produces a lot of data points that you can use to optimize your team's performance.
- Whenever there are bug issues or defects your documents are there in issue log.

Testcases :

1) Completeness
2) Clarity
3) Sufficient Data
4) Actual results
5) Environment

Software Requirements :

1) MS Project
2) Gantt Project
3) Primavera

Conclusion :

We have successfully installed Gantt Project and Completed project plan for the project .

Frequently Asked Questions :

1) What does project monitoring and controlling means ?

➔ Project Monitoring is the process of keeping a close eye on entire project management life cycle and ensure project activities are on right tasks .
➔ Controlling is an effective tool to ensure project goals and time frames.

2) What is the difference between project monitoring and project evaluation ?

➔ Monitoring is a continuous activity performed at functional level of management .
➔ Evaluation is a periodic activity , performed at business level .

3) What is Critical Path in project management ?

➔ The critical path method provides a structured approach to project scheduling, allowing you to identify the operations most important for successful project completion.
➔ The critical path (or paths) is the longest path (in time) from Start to Finish; it indicates the minimum time necessary to complete the entire project.

4) Explain the importance of Project Plan.

➔ Project planning is an important step in the overall project management :

- Ideation - The project planning is an important step to collate ideas from the customer, vendors, your team, top management, and your thoughts and put them in learning. This entails us to research further and know the gaps if there any.

- Reflect from the previous projects - The project planning is used to document and correct the ways and identify the risks and treat them so that this project does not go wary.

- Document the Project Plan - Document everything necessary such as risks, previous project failures and work around, the resources needed, time frame, cost and budget, contingency plan, etc. This ultimately helps the project for the successful completion .

5) What is Project Scheduling and explain its importance ?

➔ Scheduling in project management is the listing of activities, deliverables, and milestones within a project. A schedule also usually includes a planned start and finish date, duration, and resources assigned to each activity.
➔ Scheduling helps you keep track of everything – from scheduling staff to booking appointments, from delegating tasks to tracking deadlines, from measuring your progress to managing your workforce, and much more.