Sinhgad Institutes

Course File

Department of Computer Engineering

Subject Code	310243
Subject Name	System Programming & Operating System
Academic Year	2022-2023
Semester	IVth
Subject Teacher	Prof. K. T. Mohite

Sinhgad Technical Education Society's

SINHGAD ACADEMY OF ENGINEERING,

S. No. 40, Kondhwa (Bk), Near PMC Octroi Post,

Kondhwa –Saswad Road, Pune– 411048.

Vision

उत्तमपुरुषान् उत्तमाभियंत्रृन् निर्मातुं कटीबध्दः वयम् !

"We are committed to produce not only good engineers but good human beings, also."

Mission

"Holistic development of students and teachers is what we believe in and work for. We strive to achieve this by imbibing a unique value system, transparent work culture, excellent academic and physical environment conducive to learning, creativity and technology transfer. Our mandate is to generate, preserve and share knowledge for developing a vibrant society."

Department of Computer Engineering

Vision

"To build the Department as a Centre of Excellence for students in Computer Engineering."

Mission

"We believe in developing value based system for student and staff by providing healthy and transparent work culture to cultivate new ideas in the field of engineering and technology which will contribute to build a vibrant Society."

@TheCO-POMappingMatrix												
CO/ PO	**PO1**	**PO2**	**PO 3**	**PO4**	**PO 5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	2	1	-	-	-	-	-	-	-	-
CO 2	2	2	1	2	-	-	-	-	-	-	-	-
CO 3	2	2	1	1	-	-	-	-	-	-	-	-
CO 4	2	1	2	1	-	-	-	-	-	-	-	1
CO 5	2	2	1	2	-	-	-	-	-	-	-	1
CO 6	2	1	2	1	-	-	-	-	-	-	-	1

Sinhgad Institutes

SINHGAD ACADEMY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

SUBJECT CODE: 310248

LAB MANUAL
Laboratory Practice-I
(System Programming & Operating System)

Semester – I, Academic Year: 2021-22

SavitribaiPhulePuneUniversity

Third Year of Computer Engineering (2019 Course)310243:SystemsProgrammingandOperatingSystem

Credit:03

TeachingScheme:	Examination Scheme:Mid-
Theory:	Sem (TH) :30 MarksEnd-
03Hours/Week	Sem(TH):70Marks

Prerequisites Courses: Programming and Problem Solving (110005), Data Structures andAlgorithms (210252), Principles of Programming Languages (210255), Microprocessor (210254)**CompanionCourse:**Laboratory Practice I(310248)

CourseObjectives:

- Togetacquainted withthe basicsofSystemProgramming
- Toacquire knowledgeof data structuresusedinthe design ofSystem Software
- Tobefamiliarwiththeformatofobjectmodules,thefunctionsoflinking,relocation,andloading
- TocomprehendthestructuresandfunctionsofOperatingSystemsandprocessmanagement.
- Todealwith concurrencyand deadlockinthe OperatingSystem
- Tolearn andunderstandmemorymanagement ofOperating System

CourseOutcomes:

Oncompletion ofthe course, learnersshouldbeableto

CO1: Analyze andsynthesizebasic System Softwareanditsfunctionality.

CO2: Identifysuitable datastructures andDesign& Implement variousSystem Software**CO3:**Compare different loading schemes and analyze the performance of linker and loader**CO4:**Implementand Analyzethe performance ofprocess scheduling algorithms

CO5: Identifythemechanismtodealwithdeadlockand concurrencyissues

CO6: Demonstratememoryorganizationand memorymanagement policies

CourseContents

UnitI Introduction 08 Hours

Introduction to Systems Programming, Need of Systems Programming, Software Hierarchy,Typesof software: system software andapplication software, Machine structure.

Evolution of components of Systems Programming: Text Editors, Assembler, Macros,Compiler, Interpreter, Loader, Linker, Debugger, Device Drivers, Operating System.

Elements ofAssembly Language Programming: Assembly Language statements, Benefits of AssemblyLanguage,A simple Assembly scheme, Pass Structure of Assembler.

Design of two pass Assembler: Processing of declaration statements, Assembler Directives andimperative statements, Advanced Assembler Directives, Intermediate code forms, Pass I and PassIIof two pass Assembler.

#Exemplar/Case Studies	Studyof Debugging toolslike GDB
***MappingofCourseOutcomesfor Unit I**	CO1,CO2,CO3

UnitII MacroProcessor andCompilers 06Hours

#Exemplar/Case Studies	GNUM4Macro Processor	
***Mappingof Course OutcomesforUnit II**	CO1,CO2,CO3	
UnitIII	**Linkersand Loaders**	**07Hours**
Introduction, **Loader schemes**: Compile and Go, General Loader Scheme, Absolute Loaders,Subroutine Linkages, Relocating Loaders, Direct linking Loaders, Overlay structure, Design ofanAbsoluteLoader,DesignofDirectlinkingLoader,Self-relocatingprograms,Staticand Dynamiclinking.		
#Exemplar/Case Studies	Studytheconcepts ofClassloading inJava.	
***MappingofCourseOutcomesfor UnitIII**	CO1,CO2,CO3	
UnitIV	**OperatingSystem(OS)**	**07Hours**
Introduction: Evolution of OS, Operating System Services, Functions of Operating System.**ProcessManagement**:Process,ProcessStates:5and7statemodel,Processcontrolblock,Threads,Thread lifecycle,Multithreading Model,Process control systemcalls. **ProcessScheduling**:Uni-processorScheduling, Scheduling:Preemptive,Non-preemptive,Long-term,Medium-term,Shorttermscheduling.**SchedulingAlgorithms**:FCFS,SJF,RR,and Priority.		
#Exemplar/Case Studies	Process management in Linux /Windows/AndroidReaders-Writers problem	
***MappingofCourseOutcomesfor UnitIV**	CO4	
UnitV	**Synchronizationand ConcurrencyControl**	**07Hours**
Concurrency: Principle and issues with Concurrency, Mutual Exclusion, Hardware approach,Software approach, Semaphore, Mutex and monitor, Reader writer problem, Producer Consumerproblem,Dining Philosopher problem. **Deadlocks**:PrincipleofDeadlock,Deadlockprevention,Deadlockavoidance,Deadlock detection,Deadlockrecovery.		
#Exemplar/Case Studies	ConcurrencyMechanism:Unix/Linux/Windows.	
***Mappingof Course OutcomesforUnit V**	CO5	
UnitVI	**MemoryManagement**	**07Hours**

Introduction: MemoryManagementconcepts,MemoryManagementrequirements. **Memory Partitioning**: Fixed Partitioning, Dynamic Partitioning, Buddy Systems Fragmentation,Paging,Segmentation, Address translation. **PlacementStrategies**:FirstFit, BestFit, NextFit andWorst Fit. **Virtual Memory (VM)**: Concepts, Swapping, VM with Paging, Page Table Structure, InvertedPageTable,TranslationLookasideBuffer,PageSize,VMwithSegmentation,VMwithCombinedpaging and segmentation. **PageReplacementPolicies**:FirstInFirstOut(FIFO),LastRecentlyUsed(LRU),Optimal, Thrashing.	
#Exemplar/Case Studies	MemorymanagementinLinux/Windows/Android

***Mappingof Course OutcomesforUnit VI**	CO6

LearningResources

TextBooks:

1. JohnDonovan,"SystemsProgramming",McGrawHill,ISBN978-0--07-460482-3
2. Dhamdhere D., "Systems Programming and Operating Systems", McGraw Hill, ISBN 0 - 07 - 463579 – 4
3. Silberschatz,Galvin, Gagne,"Operating SystemPrinciples",9th Edition,Wiley, ISBN978- 1-118-06333-0

ReferenceBooks:

1. LelandBeck,"SystemSoftware: AnIntroductiontoSystemsProgramming",Pearson
2. JohnR.Levine,TonyMason,DougBrown,"Lex&Yacc",1stEdition,O'REILLY,ISBN81-7366-062-X
3. AlfredV.Aho,RaviSethi,ReffreyD.Ullman,"CompilersPrinciples,Techniques,and Tools",AddisonWesley,ISBN981-235-885-4

e-Books:

- https://www.elsevier.com/books/systems-programming/anthony/978-0-12-800729-7
- https://www.kobo.com/us/en/ebook/linux-system-programming-1
- https://www.ebooks.com/en-us/subjects/computers-operating-systems-ebooks/279/
- https://www.e-booksdirectory.com/details.php?ebook=9907

MOOCsCourses Links:

- https://www.udacity.com/course/introduction-to-operating-systems--ud923
- nptelvideolecturelink:https://nptel.ac.in/courses/106/105/106105214/
- https://www.edx.org/course/computer-hardware-and-operating-systems
- https://onlinecourses.nptel.ac.in/noc19_cs50/preview
- https://www.udemy.com/course/system-programming/

Sinhgad College of Engineering
Department of ComputerEngineering

310248: LaboratoryPractice-I

TeachingScheme:
Practical:4Hrs/Week
Credits:02

Examination Scheme:
Term work: 25Marks
Practical: 25Marks

List of Laboratory Assignments

Instructions

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, punctuality and

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus. For the elective subjects students should form group of 3-4 students. The faculty coordinator will take care that all the assignment should be assigned to class and minimum two assignments are compulsory for each group.

Programming tools recommended: -

Human computer Interface-GUI in python

Internet of Things and Embedded System- Raspberry Pi/Arduino Programming; Arduino IDE/Python Interfacing. Other IoT devices

Software project management-MS project/Gantt Project/Primavera

Course Objectives:

1. To learn system programmingtools
2. To learn modern operatingsystem

Course Outcome:

On completion of this course, learners will be able to

CO1: Implement different system software's like assembler, macro processor, DLL, etc.

CO2: Implement concept of synchronization and concurrency

CO3: Implement scheduling policies and memory management concepts of operating system

Assignment No.:01

ProblemStatement: Designsuitabledatastructuresandimplementpass-Iofatwo-pass assembler for pseudo machine in Java/C++ using object oriented feature. Implementation should consist of a few instructions from each category and few assembler directives.

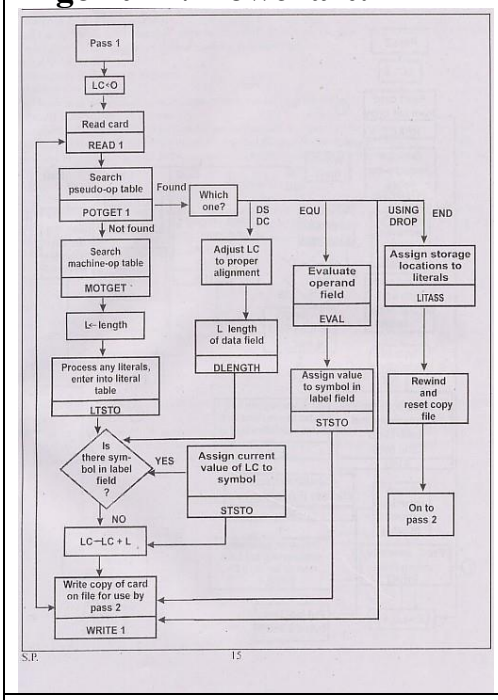
Objectives:

1. To study the design and implementation of 1st pass of two passassembler.
2. To study the categorized instruction set ofassembler.
3. To study the data structure used in assemblerimplementation.

Theory:

1. Explain various Data and Instruction formats of assembly languageprogramming.
2. Explain the design of Pass- I of assembler with the help of flowchart andexample.
3. Discuss various Data structure used in Pass-I along with its format and significance of each field.

Algorithm/Flowchart:



Design diagrams (if any):

1. ClassDiagram
2. Use caseDiagram
3. ERDiagram

Input:

Source code of Assembly Language

```
SAMPLE    START     100
          USING     *, 15
          L         1,FOUR
```

```
                    A       1,=F'3'
                    ST      1,RESULT
                    SR      1, 2
                    LTORG
                    L       2,FIVE
                    A       2,=F'5'
                    A       2,=F'7'
FIVE                DC      F'5'
FOUR                DC      F'4'
RESULT              DS      1F
                    END
```

Output:

```
100     SAMPLE      START       100
100                 USING       *, 15
100                 L           1,FOUR
104                 A           1,=F'3'
108                 ST          1,RESULT
112                 SR          1, 2
114                 LTORG
124                 L           2,FIVE
128                 A           2,=F'5'
132                 A           2,=F'7'
136     FIVE        DC          F'5'
140     FOUR        DC          F'4'
144     RESULT      DS          1F
152                 5
156                 7
160                 END
```

Machine Opcode Table (MOT)

Mnemonic	Hex/ Binary Code	Length (Bytes)	Format
L	5A	4	RX
A	1B	4	RX
ST	50	4	RX
SR	18	2	RR

Pseudo Opcode Table (POT)

Pseudo op	Address / Name of Procedure to implement pseudo operation
START	PSTART
USING	PUSING
DC	PDC
DS	PDS
LTORG	PLTORG
END	PEND

Symbol Table (ST)

Sr.No	Symbolname	Address	Value	Length	Relocation
1	SAMPLE	100	--	160	R
2	FIVE	136	5	4	R
3	FOUR	140	4	4	R
4	RESULT	144	___	4	R

Literal Table (LT)

Sr. No	Literal	Address	Length
1	3	120	4
2	5	152	4
3	7	156	4

Instructions :

Not specific

Test Cases:

1. Check syntax of instruction (Correct andwrong)
2. Symbol notfound
3. Wronginstruction
4. Duplicate symboldeclaration
5. Test the output of program by changing value of START pseudoopcode.
6. Test the output of program by changing position of LTORGpseudo-op.

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement:

Not specific

Frequently Asked Questions:

1. What is two passassembler?
2. What is the significance of symbol table?
3. Explain the assembler directives EQU,ORIGIN.
4. Explain the assembler directives START, END,LTORG.
5. What is the use of POOLTAB and LITTAB?
6. How literals are handled in passI?
7. What are the tasks done in Pass I?
8. How error handling is done in passI?
9. Which intermediate data structures are designed and implemented inPassI?
10. What is the format of a machine code generated inPassII?
11. What is forward reference? How it is resolved byassembler?
12. How error handling is done in pass II?
13. What is the difference between IS, DL andAD?

14. What are the tasks done in Pass II?

Conclusion:

Input assembly language program is processed by applying Pass-I algorithm of assembler and intermediate data structures, Symbol Table, Literal Table, MOT, POT, BT, etc. are generated.

Assignment No.:02

Problem Statement:

Implement Pass-II of two pass assembler for pseudo-machine in Java/C++ using object oriented features. The output of assignment-1 (intermediate file and symbol table) should be input for this assignment.

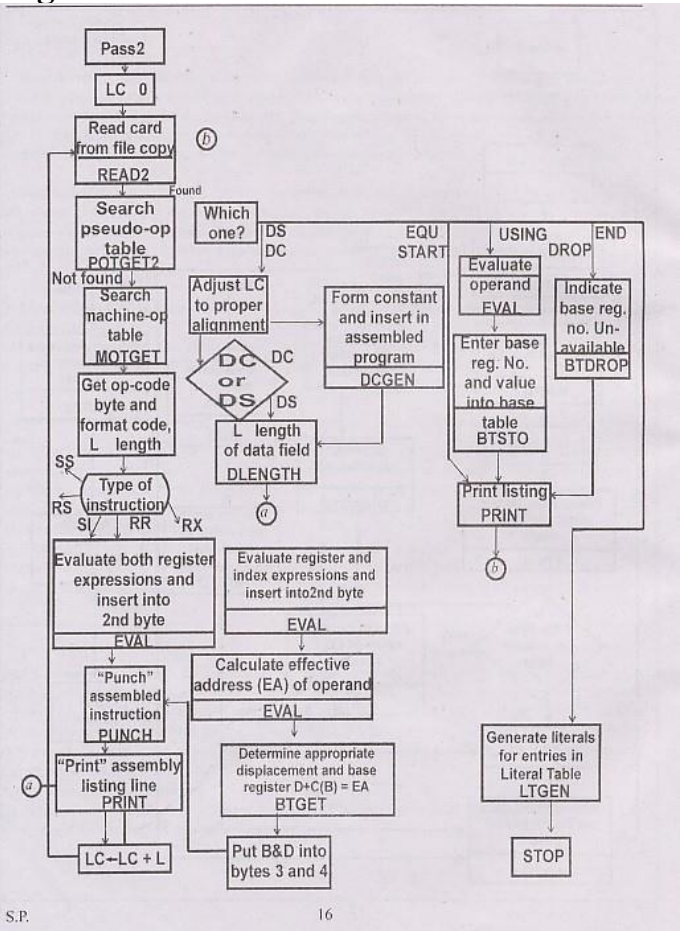
Objectives:

1. To study the design and implementation of 2nd pass of two passassembler.
2. To study the data structure used in Pass-2 of assemblerimplementation.

Theory:

1. Explain the design of Pass- II of assembler with the help of flowchart and example.

Algorithm/Flowchart:



Design diagrams (if any):

1. ClassDiagram
2. Use caseDiagram
3. ERDiagram

Input:

Intermediate code of pass-1.

LC LABEL INSTR. OPERANDS

```
-------------------------------------------------------------
100     SAMPLE      START       100
100                 USING       *, 15
100                 L           1,FOUR
104                 A           1,=F'3'
108                 ST          1,RESULT
112                 SR          1, 2
114                 LTORG
124                 L           2,FIVE
128                 A           2,=F'5'
132                 A           2,=F'7'
136     FIVE        DC          F'5'
140     FOUR        DC          F'4'
144     RESULT      DS          1F
152                 5
156                 7
160                 END
```

Machine Opcode Table (MOT)

Mnemonic	Hex / Binary Code	Length (Bytes)	Format
L	5A	4	RX
A	1B	4	RX
ST	50	4	RX
SR	18	2	RR

Pseudo Opcode Table (POT)

Pseudo op	Address / Name of Procedure to implement pseudo operation
START	PSTART
USING	PUSING
DC	PDC
DS	PDS
LTORG	PLTORG
END	PEND

Symbol Table (ST)

Sr. No	Symbol name	Address	Value	Length	Relocation
1	SAMPLE	100	--	160	R
2	FIVE	136	5	4	R
3	FOUR	140	4	4	R
4	RESULT	144	___	4	R

Literal Table

(LT)Sr.No		Address	Length
	Literal	120	4
1	3	152	4
2	5	156	4
3	7		

Output:

Base Table (BT)

Register no	Availability	Value/ Contents
1	N	--
:	:	:
:	:	:
:	:	:
15	Y	100

Object Code

LC	OPCODE	OPERAND
100	5A	1,40(0,15)
104	1B	1,20(0,15)
108	50	1,44(0,15)
112	18	1,2
124	5A	2,36(0,15)
128	1B	2,52(0,15)
132	1B	2,56(0,15)

Instructions :

1.
2.
3.

Test Cases:

1. Check syntax of instruction (Correct andwrong)
2. Symbol notfound
3. Wronginstruction
4. Duplicate symboldeclaration
5. Test the output of program by changing value of START & USING pseudoopcode.

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement:

Frequently Asked Questions:

1. What is two passassembler?
2. What is the significance of symbol table?
3. Explain the assembler directives EQU,ORIGIN.
4. Explain the assembler directives START, END,LTORG.
5. What is the use of POOLTAB and LITTAB?
6. How literals are handled in pass I?
7. What are the tasks done in Pass I?
8. How error handling is done in passI?
9. Which intermediate data structures are designed and implemented inPassI?
10. What is the format of a machine code generated inPassII?
11. What is forward reference? How it is resolved byassembler?
12. How error handling is done in pass II?
13. What is the difference between IS, DL andAD?

Conclusion:

The intermediate data structures generated in Pass-I of assembler are given as input to the Pass-II of assembler, processed by applying Pass-II algorithm of assembler and machine code is generated

Assignment No.:03

Problem Statement: Design suitable data structures and implement Pass-I of a two pass macro processor using OOP features in Java/C++. The output of Pass-I (MNT, MDT, ALA & Intermediate code file without any macro definitions) should be input for Pass-II.

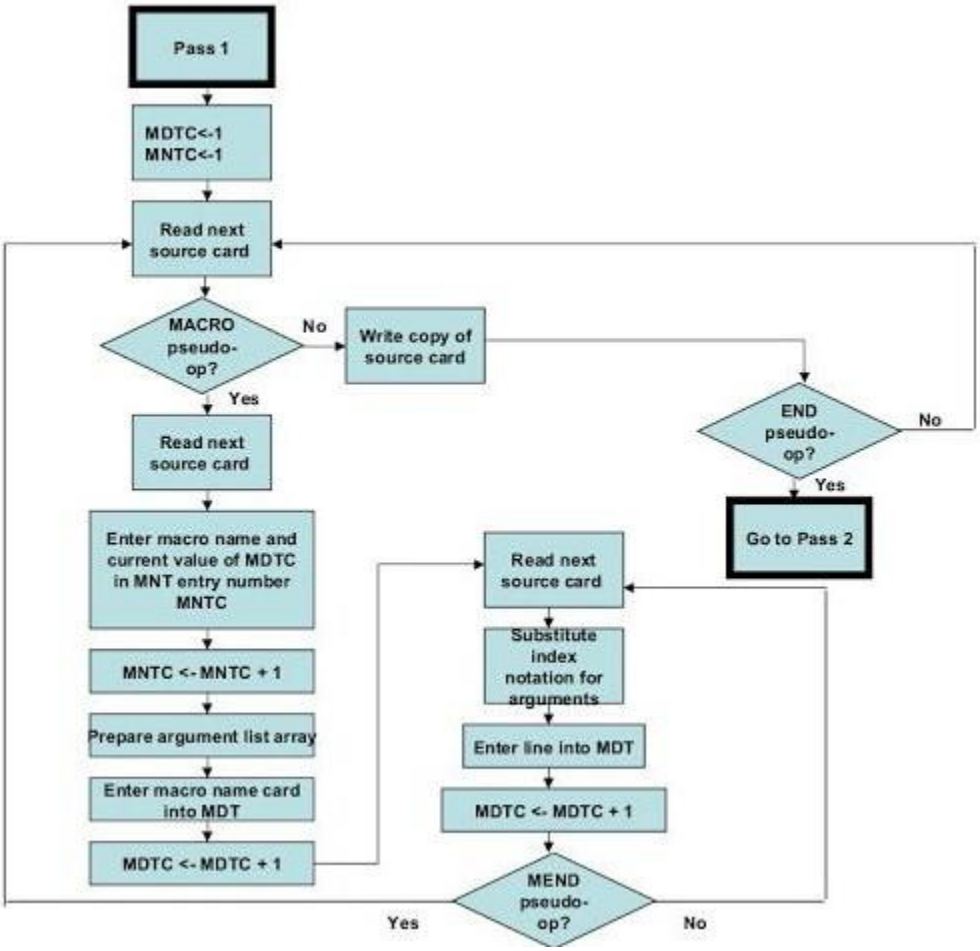
Objectives:

1. To identify and design different data structure used inmacro-processor implementation
2. To apply knowledge in implementation of two passmicroprocessor.

Theory:

1. What is macroprocessor?
2. Differentiate Macro andFunction?
3. Explain the design of Pass- I of macro-processor with the help offlowchart?
4. Explain the design of Data structure used inPass-I?
5. Explain the data structures used inPass-I?

Algorithm/Flowchart:



Design diagrams (if any):

1. Classdiagram
2. Sequence diagram
3.

Input:

Small assembly language program with macros written in file input.asm.

```
        MACRO
&lab    ADDS &arg1,&arg2
&lab    L 1,&arg1
        A 1, &arg2
        MEND
PROG START 0
        BALR 15,0
        USING *,15
LAB ADDS DATA1, DATA2
        ST 4,1
DATA1 DC F'3'
DATA2 DC F'4'
        END
```

Output:

Assembly language program without macro definition but with macro call.

Note: Follow the following templates during implementation

Macro Name Table (MNT) :

Index	Macro Name	MDT Index
1	ADDS	15

Macro Definition Table (MDT) :

Index	Macro Definition Card entry		
15	&lab	ADDS	&arg1, &arg2
16	#0	L	1, #1
17		A	1, #2
18		MEND	

Argument List Array (ALA) :

Index	Arguments
0	&lab
1	&agr1
2	&arg2

Instructions :

1.
2.
3.

Test Cases:

1. Check macro end not found.
2. Duplicate macro name found.
3. Check program output by changing macro name and parameterlist.
4. Handle label in macrodefinition.
5. Handle multiple macro definitions andcalls

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement: N/A

Frequently Asked Questions:

1. Definemacro?
2. Define purpose of pass-1 of two pass macroprocessor
3. List out types of macroarguments
4. What is the use of MDT-index field inMNT?
5. What we store inALA?

Conclusion: We have successfully completed implementation of Pass-I of macro processor.

Assignment No.:04

Problem Statement: Design suitable data structures and implement Pass-II of a two pass macro processor using OOP features in Java/C++. The output of Pass-I (MNT, MDT, ALA & Intermediate code file without any macro definitions) should be input for Pass-II.

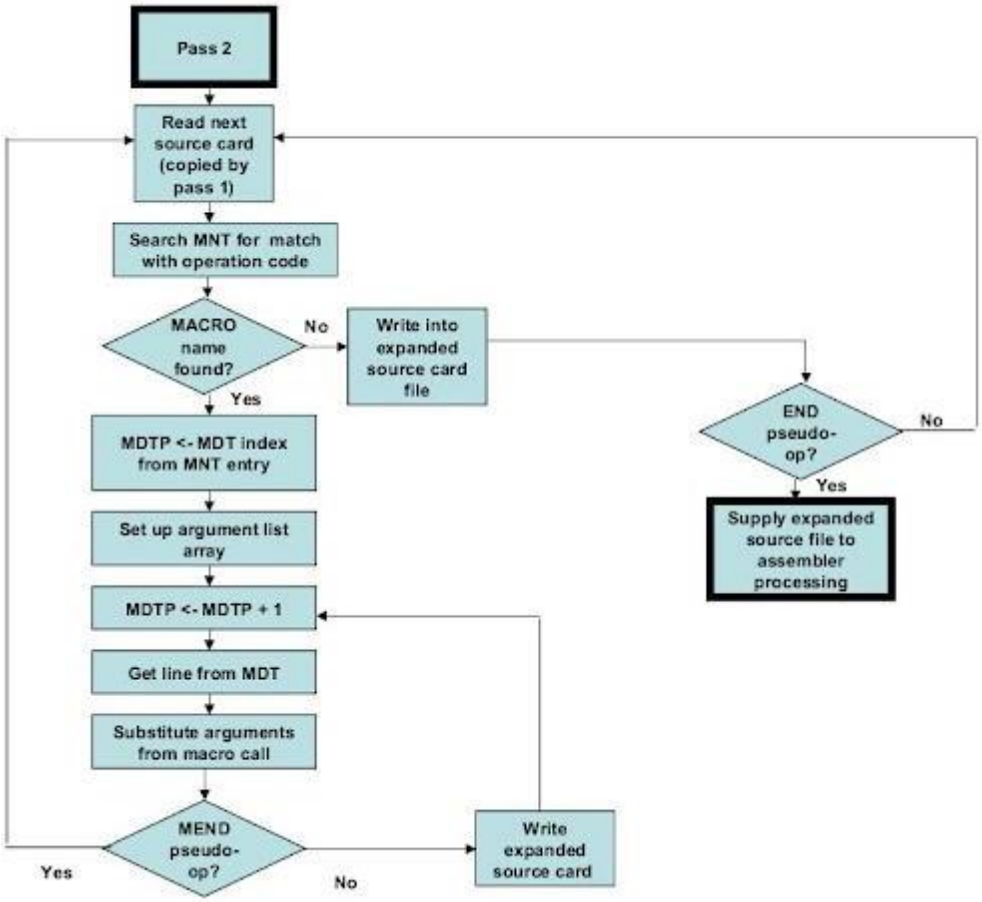
Objectives:

1. To identify and design different data structure used in macro-processorimplementation
2. To apply knowledge in implementation of pass-2 of two passmicroprocessor.

Theory:

1. Explain design steps of two pass microprocessor, types of statements, data structuresrequired and flowcharts.

Algorithm/Flowchart:



Design diagrams (if any):

1. Classdiagram
2. Sequence diagram
3.

Input: Output of pass-1 (Intermediate File) given as a input to pass-2.

```
PROG START 0
BALR 15,0
USING *,15
LAB ADDS DATA1, DATA2
ST 4,1
DATA1 DC F'3'
DATA2 DC F'4'
END
```

Output:

Assembly language program without macro definition and macro call.

```
PROG START 0
BALR 15,0
USING*,15
LAB L 1, DATA1
A 1,DATA2
ST 4,1
DATA1 DC F'3'
DATA2 DC F'4'
END
```

Macro Name Table (MNT):

Index	Macro Definition Card entry		
15	&lab	ADDS	&arg1, &arg2
16	#0	L	1, #1
17		A	1, #2
18		MEND	

Macro Definition Table (MDT):

Index	Macro Name	MDT Index
1	ADDS	15

Argument List Array (ALA) :

Index	Arguments
0	LAB
1	DATA2
2	DATA3

Instructions :

1.
2.
3.

Test Cases:

1. Check macro definition notfound.
2. Check program output by changing parameter list in macrocall.

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement: N/A

Frequently Asked Questions:

1. What is macroexpansion?
2. Define purpose of pass-2 of two pass macroprocessor
3. What is positionalarguments?
4. What is the use of MDT-index field inMNT?
5. What is the use of MNT table while processing macrocall?

Conclusion: We have successfully completed implementation of Pass-II of macro processor.

Assignment No.:05

Problem Statement:

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

Objectives:

1. To study and understand concept ofDLL
2. To understand JNI
3. To implement DLL usingJNI

Theory:

1. What is DLL? Significance of DLL. Advantages/ Disadvantages ofDLL
2. What is Native Interface? Reasons to useJNI.
3. What is sharedobject?

Algorithm/Flowchart:

1. Write a Java Class that uses C Codes -TestJNI.java

```
public class TestJNI {
static {
System.loadLibrary(“cal"); // Load native library at runtime
// cal.dll (Windows) or libcal.so (Unix)
}
// Declare a native method add() that receives nothing and returns void private native int add
(int n1,int n2);
// Test Driver
public static void main(String[] args) {
// invoke the native method
System.out.println(“Addition is=”+new TestJNI().add(10,20);
} }
```

Compile Java code:

```
javac TestJNI.java
```

2. Create the C/C++ Header file -TestJNI.h

```
javah -jni TestJNI
```

3. C Implementation -TestJNI.c

```
#include <jni.h>
#include <stdio.h>
#include “TestJNI.h"
// Implementation of native method add() of TestJNI class
```

```
JNIEXPORT jint JNICALL Java_TestJNI_add(JNIEnv *env, jobject thisObj,jint n1,jint n2)
{
jint res;
res=n1+n2;
return res;
}
```

Compile c-program:

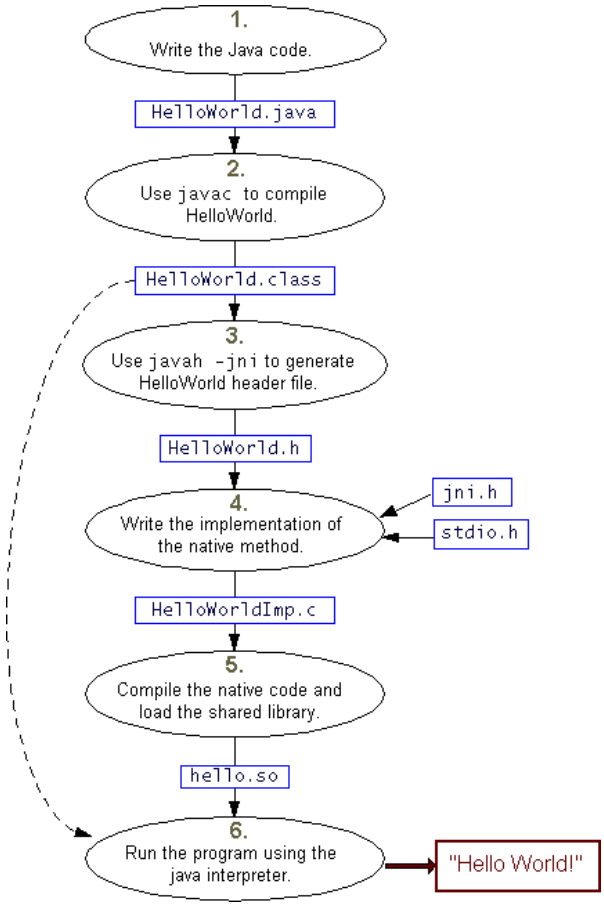
```
$gcc -I /usr/local/jdk1.8.0_91/include /usr/local/jdk1.8.0_91/include/linux -o libcal.so
–shared TestJNI.c
```

4. Run java program

```
$java -Djava.library.path=. TestJNI
Addition is=30
```

5. Repeat step 1-4 for all mathematical operations mentioned in problemstatement.

Flowchart:



Design diagrams (if any):

1. Use CaseDiagram
2. SequenceDiagram

Input:

1. n1=20 2. n2=10
Output: 1.Addition=30
Instructions : 1. This assignment can be implemented using VB application and C++ DLL using visual studio on windows
Test Cases: 1. Divide byzero 2. Missingarguments
Software Requirement: 1. Fedora 2. Jdk 3. Eclipse/ equivalentIDE
Hardware Requirement:
Frequently Asked Questions: 1. Difference between static link library and dynamic linklibrary 2. What is sharedobject? 3. Advantages/Disadvantages of usingJNI
Conclusion: Successfully implemented DLL and tested it with java application

Assignment No.:06

Problem Statement:

Write a program to simulate CPU Scheduling Algorithms: FCFS, SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive).

Objectives:

1. To study the process management and various scheduling policies viz. Preemptive andNon preemptive.
2. To study and analyze different schedulingalgorithms.

Theory :

1. Define process. Explain need of processscheduling.
2. Explain different scheduling criteria and policies for schedulingprocesses.
3. Explain possible processstates
4. Explain FCFS, SJF(Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive) and determine waiting time, turnaround time, throughput using eachalgorithm.

Algorithm/Flowchart:

1. FCFS

1. Input the processes along with their burst time(bt).
2. Find waiting time (wt) for allprocesses.
3. As first process that comes need not to wait so waiting time for process 1 will be 0 i.e. wt[0] =0.
4. Find **waiting time** for all other processes i.e. for process i -> wt[i] = bt[i-1] + wt[i-1].
5. Find **turnaround time** = waiting_time + burst_time for allprocesses.
6. Find **average waiting time** = total_waiting_time /no_of_processes.
7. Similarly,find**averageturnaroundtime**=total_turn_around_time/no_of_processes.

1. SJF

1. Traverse until all process gets completelyexecuted.

a) Find process with minimum remaining time at every single timelap.

b) Reduce its time by1.

c) Check if its remaining time becomes0

d) Increment the counter of processcompletion.

e) Completion time of current process = current_time+1;

e) Calculate waiting time for each completed process.

wt[i]= Completion time -arrival_time-burst_time

f) Increment time lap byone.

2. Find turnaround time(waiting_time+burst_time).

3. Priority

1- First input the processes with their burst time andpriority.

2- Sort the processes, burst time and priority according to the priority.

3- Now simply apply FCFSalgorithm.

4. RR

1- Create an array **rem_bt[]** to keep track of remaining burst time of processes. This array is initially a copy of bt[] (burst timesarray)
2- Create another array **wt[]** to store waiting times of processes. Initialize this array as 0.
3- Initialize time : t =0
4- Keep traversing the all processes while all processes are not done. Do following for i'th process if it is not doneyet.
a- If rem_bt[i] >quantum
(i) t = t +quantum
(ii) bt_rem[i] -=quantum;
c- Else // Last cycle for thisprocess
(i) t = t +bt_rem[i];
(ii) wt[i] = t -bt[i]
(ii) bt_rem[i] = 0; // This process is over

Design diagrams (if any):

Class Diagram
Use Case Diagram
SequenceDiagram

Input:

1. Enter the number ofprocesses
2. Enter burst time and arrival time of eachprocess

Output:

1. Compute Waiting time, turnaround time, average waiting time, average turnaround time and throughput.

For each algorithm display result as follows:

Process	Burst Time	Arrival Time	Waiting Time	Turnaround Time
P1				
P2				
P3				
-				

Calculate

1. Average waitingtime=
2. Average turnaroundtime=
3. Throughput=

Instructions :
1.
2.
3.

Test Cases:

1. Check arrival time of all process should not be same.

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement: for simulation no dependency

Frequently Asked Questions:

1. What are the types of CPUscheduler?
2. What is the difference between long and short termscheduling?
3. Logic ofprogram?
4. What is preemptive and non-preemptivescheduling?
5. What are types of schedulingalgorithms?
6. Why Priority scheduling may cause low-priority processes tostarve?
7. What are the goals ofscheduling?
8. Define the difference between preemptive and non-preemptivescheduling.
9. Which scheduling algorithm is best?Why?

Conclusion:
CPU policies implemented successfully

Assignment No.:7

Problem Statement:

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

Objectives:

1. To understand reader writer synchronizationproblem
2. To solve reader-writer synchronization problem using mutex andsemaphore

Theory:

- There is a data area shared among a number of processorregisters.
- The data area could be a file, a block of main memory, or even a bank ofprocessor 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 satisfiedare
  - Any number of readers may read simultaneously read thefile.
  - Only one write at a time may write to thefile.
  - If a writer is writing to the file, no reader may readit.

Semaphore:

Definition: Semaphores are system variables used for synchronization of process

Two types of Semaphore:

- **Counting semaphore –** integer value can range over an unrestricteddomain
- **Binary semaphore–**
  - Integer value can range only between 0 and 1; can be simpler toimplement
  - Also known as mutexlocks

Semaphore functions:

Package: import java.util.concurrent.Semaphore;

1) **To initialize asemaphore:**

   Semaphore Sem1 = new Semaphore(1);

2) **To wait on asemaphore:**

```
/* Wait (S)
  whileS<=0
        no-op;
  S - -;
*/
Sem1.acquire();
```

3) **To signal on asemaphore:**

```
/* Signal(S)
    S ++;

*/
mutex.release();
```

Algorithm/Flowchart:

Algorithm for Reader Writer:

1. **import java.util.concurrent.Semaphore;**
2. **Create a classRW**
3. **Declare semaphores – mutex andwrt**
4. **Declare integer variable readcount =0**
5. **Create a nested class Reader implementsRunnable**
    a. **Override run method (ReaderLogic)**
        i. wait(mutex);
        ii. readcount := readcount+1;
        iii. if readcount = 1then
        iv. wait(wrt);
        v. signal(mutex);
        vi. …
        vii. reading isperformed
        viii. …
        ix. wait(mutex);
        x. readcount := readcount –1;
        xi. if readcount = 0 thensignal(wrt);
        xii. signal(mutex):
6. **Create a nested class Writer implementsRunnable**
    a. **Override run method (WriterLogic)**
        i. wait(wrt);
        ii. …
        iii. writing isperformed
        iv. …
        v. signal(wrt);
7. **Create a class main**
    a. **Create Threads for Reader andWriter**
    b. **Start these thread**

Design diagrams (if any):

Input:
1. Number ofReaders
2. Number ofWriters

Output:
1. Execution of Readers and Writers

Instructions:
1.
2.
3.

Test Cases:

1. Create 5 readers first and then 5 writers and check their sequence ofexecution
2. Create 5 writers first and then 5 readers and check their sequence ofexecution
3. Create 5 writers and 5 readers alternatively and check their sequence ofexecution

Software Requirement:

1. Java
2. Eclipse/NetBeans

Hardware Requirement:

1. Nothing Special

Frequently Asked Questions:

1. What is synchronization ofthreads?
2. Explain reader writer problem
3. Explain wait and sequencefunctions
4. What issemaphore.
5. What are different types ofsemaphore

Conclusion: Implemented Reader Writer synchronization problem using semaphores in Java

Assignment No.:08

Problem Statement:

Write a Java/C++ program to simulate memory placement strategies

1. FirstFit
2. BestFit
3. Worst Fit
4. Next Fit

Objectives:

1. To acquire knowledge memory placementstrategies
2. To be able to implement memory placementstrategies

Theory:

1. Why we need memory placementstrategies?
2. What isfragmentation?
3. Explain working of memory placement strategies with suitableexample.

Algorithm/Flowchart:

1. First Fit algorithm/pseudo code

- Read all requiredinput
- FOR i<-0 to all jobs‘js’
  - FOR j<-0 to all blocks‘bs’
    - IFblock[j]>=jobs[i]
      - Check j^{th} block is already in use orfree
        - Continue and search next freeblock
      - Otherwise allocate j^{th} block to i^{th}job
- Display all job with allocated blocks andfragmentation

2. First Fit algorithm/pseudo code

- Read all requiredinput
- FOR i<-0 to all jobs‘js’
  - SET BestInd←-1
  - FOR j<-0 to all blocks‘bs’
    - IFblock[j]>=jobs[i]
      - IF Block is free and BestInd==-1 THENSET BestInd←j
      - ELSEIF Block is free and block[BestInd]>block[j] THEN SETBestInd←j
      - ELSE continue with nextblock
        - Continue and search next freeblock

- IF BestInd!=-1 THEN allocate j^{th} block to i^{th}job
- Display all job with allocated blocks and fragmentation

3. Worst Fit Algorithm/Pseudo code

- Read all requiredinput
- FOR i<-0 to all jobs'js'
  - SET WstInd←-1
  - FOR j<-0 to all blocks'bs'
    - IFblock[j]>=jobs[i]
      - IF Block is free and WstInd==-1 THEN SET WstInd←j
      - ELSEIF Block is free and block[WstInd]<block[j] THEN SETWstInd←j
      - ELSE continue with nextblock
        - Continue and search next freeblock
      - IF WstInd!=-1 THEN allocate j^{th} block to i^{th}job
- Display all job with allocated blocks andfragmentation

4. As above write algorithm of Next Fitstrategies

Design diagrams (if any):

1. Class diagram

Input:

- No. of jobs (js) & No. of blocks(bs)
- Job size of all jobs & Block size of allblocks

ForExample:

```
js=4
bs=5
block[]  = {100, 500, 200, 300,600};
jobs[] = {212, 417, 112, 426};
```

Output:

Sample output of Worst Fit algorithm (same way generate o/p for other algorithms)-

ProcessNo.	ProcessSize	Blockno.
1	212	5
2	417	2

3	112	4
4	426	NotAllocated

Instructions : not specific

Test Cases:

Software Requirement:
1. Eclipse IDE
2. Java

Hardware Requirement: Not specific

Frequently Asked Questions:
1. Which algorithm is best andwhy?
2. Need of allocating blocks tojobs?
3. What is the time taken by each algorithm forexecution?

Conclusion: successfully implemented simulation of memory placement strategies.

Assignment No.:09

Problem Statement:

Write a Java Program (using OOP features) to implement paging simulation using

1. FIFO
2. Least Recently Used(LRU)
3. Optimalalgorithm

Objectives:

1. To study page replacement policies to understand memorymanagement.
2. To understand efficient frame management using replacementpolicies.

Theory:

CONCEPT OF PAGE REPLACEMENT:

1. Page Fault: Absence of page when referenced in main memory during paging leads to a page fault.
2. Page Replacement: Replacement of already existing page from main memory by therequired new page is called as page replacement. And the techniques used for it are called as page replacementalgorithms.

NEED OF PAGE REPLACEMENT:

Page replacement is used primarily for the virtual memory management because in virtual memory paging system principal issue is replacement i.e. which page is to be removed so as to bring in the new page, thus the use of the page replacement algorithms. Demand paging is the technique used to increase system throughput. To implement demand paging page replacement is primaryrequirement.Ifasystemhasbetterpagereplacementtechniqueitimprovesdemandpaging which in turn drastically yields system performancegains.

PAGE REPLACEMENT POLICIES:

1. Determine which page to be removed from main memory.
2. Find a free frame.
   1) If a frame is found use it
   2) if no free frame found, use page replacement algorithm to select avictim frame.
   3) Write the victim page to thedisk.
3. Read the desired page into the new free frame, change the page and frametables.
4. Restart the userprocess.

PAGE REPLACEMENT ALGORITHMS:

1. **FIFO**

   This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal.

2. **OPTIMAL PAGE REPLACEMENT ALGORITHM:** Replace the page that will not be usedforlongestperiodoftimeascomparedtotheotherpagesinmainmemory.Anoptimal page replacement algorithm has lowest page fault rate of all algorithm. It is called as OPT orMIN.

   ADVANTAGE:

   1) This algorithm guarantees the lowest possible page-fault rate for a fixed no.of frames.

   DISADVANTAGE:

   1) The optimal page replacement algorithm is very difficult to implement, as it requires the knowledge of reference strings i.e. strings of memory references.

3. **LEAST RECENTLY USED (LRU):** LRU algorithm uses the time of the page's last usage. It uses the recent past as an approximation of the near future, then we can replace thepagethathasnotbeenusedforthelongestperiodofthetimei.e.thepagehavinglarger idle time isreplaced.

   ADVANTAGE:

   1) The LRU policy is often used for page replacement and is considered tobe good.

   DISADVANTAGES:

   1) It is very difficult toimplement.
   2) Requires substantial hardwareassistance.
   3) The problematic determination of the order for the frames defined by thetime of lastusage

Algorithm/Flowchart:

1. FIFO :

1. Start theprocess

2. Read number of pagesn

3. Read number of pagesno

4. Read page numbers into an arraya[i]

5. Initialize avail[i]=0 .to check pagehit

6. Replace the page with circular queue, while re-placing check page availability in theframe

Place avail[i]=1 if page is placed in the frame Count pagefaults

7. Print theresults.

8. Stop theprocess.

2. **LEAST RECENTLY USED**

1.Start theprocess

2. Declare thesize

3. Get the number of pages to beinserted

4. Get thevalue

5. Declare counter andstack

6. Select the least recently used page by countervalue

7. Stack them according theselection.

8. Display thevalues

9. Stop theprocess

3. **OPTIMAL**

ALGORTHIM:

1. StartProgram

2. Read Number Of Pages And Frames

3.Read Each PageValue

4. Search For Page In TheFrames

5. If Not Available Allocate Free Frame

6. If No Frames Is Free Replace The Page With The Page That Is Least Used

7.Print Page Number Of PageFaults

8.Stop process.

Design diagrams (if any):

1. Class Diagram

Input:

1. Number offrames
2. Number ofpages

3. Page sequence

Output:

1. Sequence of allocation of pages in frames (for eachalgorithm)
2. Cache hit and cache missratio.

Instructions :

Test Cases:

1. Test the page hit and miss ratio for different size of pageframes.
2. Test the page hit and miss ratio for both algorithms with different pagesequences.

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement:

Frequently Asked Questions:

1. What is virtualmemory?
2. Explain working of LRU page replacementalgorithm
3. Explain working of OPTIMAL page replacementalgorithm
4. Which Page replacement algorithm is best?
5. Explain what is Belody'sAnomaly?
6. Explain the scenario in which page replacement algorithm isused?
7. Explain what is pagefault?
8. Explain what is pagingscheme?
9. Explain what is counting based page replacementalgorithms?

Conclusion: Successfully implemented all page replacement policies.