Sinhgad Institutes | Date

Assignment 3

Aim: Write a program to create Dynamic Link library for any mathematical operations (arithmatic, trignometric and string operations) and write an application program to test it (Java Native Interface / use VB / VC++)

objectives:
1. To study and understand concept of DLL
2. To understand JNI
3. To implement DLL using JNI

Theory:

1. What is DLL ? Significance of DLL. Advantage/ Disadvantages of DLL.

→

Dynamic Link Library :

A dynamic link library (DLL) is a collection of small programs that can be looded when needed by larger programs and used at the some time. The small program lets the larger program comm-unicate with a specific device, such as a printer or scanner. It is often packaged as a DLL program, which is usually reffered to as a DLL file. DLL file that support specific device operations 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 that using being compiled into the main program.

Sinhgad Institutes

Date

Significance of DLL:

- DLL 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 required an EXE for the operating system to load it through an entry point, hence, the existing 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.

- DLL provides 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 Window and os/2 can be thought of as a collection of DLL that are upgraded.

Advantages-

- It saves memory and reduces swapping Many process 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

copy of the DLL on disk.

- Upgrades to the DLL are easier.

- 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.

Disadvantages -

- A potential disadvantages of using DLLs is that the application is not self-contained, it depends on the existence of a seperate DLL module.

2. What is Native Interface ? Reason to use JNI.

→ In software design, the Java Native Interface (JNI) is a foreign function interface programing framework that ~~ne~~ enables JAVA code running in a JAVA virtual machine (JVM) to call and be called by native applications (program specific to a hardware and operating system platform) and libraries written in other languages such as C, C++, and assembly.

JNI enables programmers to write native method to handle situations when an application cannot be written entirely in the JAVA programming language, e.g when the standard JAVA class library does not support platform-specific feature of program library. It is also used to modify an existing application to be accessible to JAVA application. Many of the standard library classes depends on JNI to provide functionality to the user developer.

Sinhgad Institutes | Date

3. What is sharedobject?

→ A shared object is an indivisible unit that is generated form one or more relocatable objets. shared objects can be bound with dynamic executable to form a ~~rna~~ runable process. As their name implies. shared object can be shared by more than one applications. Because of this potentially far-reaching effect, this chapter describes this form of link-editor output in greater depth that has been covered in previous chapters

for a shared object to be bound to a dynamic executable or another shared object, it must first be available to the link-edit of the required output file during this link. ~~end~~. edit, any input shared objects are interpreted as if they had been added to logical address space of the output file begin produced. All the functionality of the shared object is made available to the output file.

Design diagram (if any):

1. use case Diagram
2. sequence diagram.

Input 1. n1 = 20, e. n2 = 10

output 1. Addition = 30

Instruction :

1. This assignment can be implemented using VB applications and c++ DLL using visual studio on windows.

Sinhgad Institutes | Date

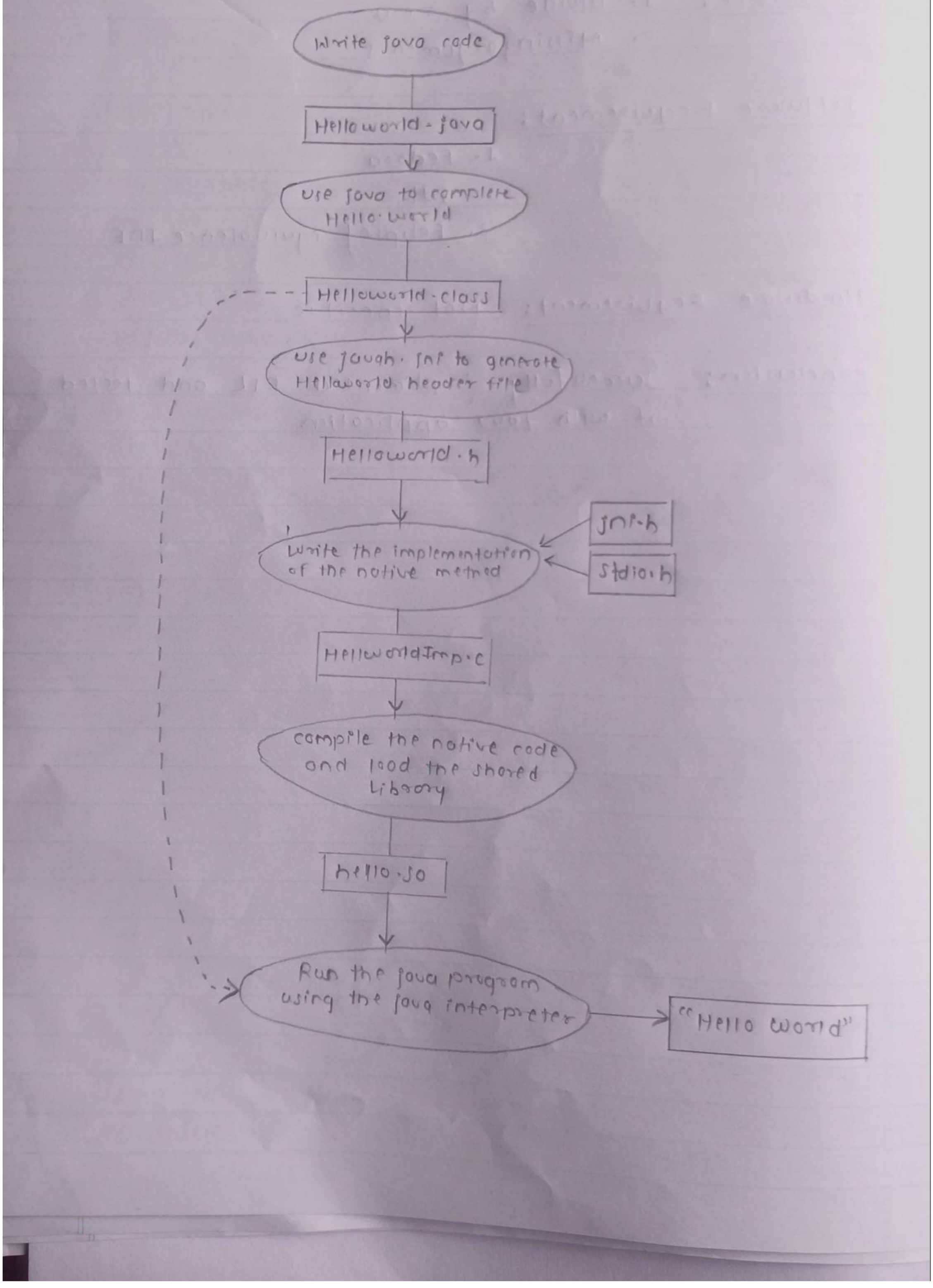
Test case: 1. Divide by zero
2. Missingorguments

Software Requirement:
1. Fedora
2. Jdk
3. Eclipse/ equivalence IDE

Hardware Requirement: Not specific

conclusion: successfully implemented DLL and tested it with Java application

Flowchart -



Sinhgad Institutes | Date

Assignment 4

Aim: Write a program to simulate CPU scheduling Algorithms: FCFS, SJF (premptive), priority (Non-preemptive) and Round Robin (preemptive).

objective:
1. To study the process management and various scheduling policies viz. preemptive and Non-preemptive.
2. To study and analyze different schedulingalgorithm.

Theory: 1. A process is basically a program in execution The execution of a process must progress in a sequential fashion. A process is defined as an entity which represents the basic unit of work to be implemented in the system or put it in simple terms, we write our computer programs in a text file and when we write our computer programs in a text file and when we execute this program, it becomes a process which perform.

The process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy Process scheduling is an essential part of a multiprog-ramming operating system. Such operating system allow more than one process to be loaded into the executable memory at a time and the loaded proce shares the CPU using time multiplexing.

Siddhgad Institutes | Date

2. Explain different scheduling criteria and policies for scheduling process.

— The scheduling criteria includes the following.

A) CPU utilization-

We want to keep the CPU as busy as possible, CPU utilization may range from 0 to 100 present. In a real system, it should range from 40 percent to 90 percent.

B) Throughput-

If the CPU is busy executing processes, then work is being done. One measure of work is the number of processes completed per time unit called throughput.

C) Turnaround time-

From the point of view of a particular process, the important criterion is how long it takes to execute that process. The internal from the time of submission of a process to the time of completion is the turnaround time.

D) Waiting time-

The CPU scheduling algorithm does not affect the amount of time during which a process executes or does I/O; it affects only the amount of time that a process spends waiting in the ready queue.

E) Response time-

In an interactive, system turnaround time may not be the best criterion. Often a process can produce some output fairly early, and can continue computing new result while previous results are being output to users.

Sinhgad Institutes | Date

3. Explain possible processstates.

→ A process is a program in execution and it is more that a program code called as text section and this concept works under all the operating system because all the task perform by the operating system need a process to perform the task

The process executes when it changes the state. The state of a process is defined by the current activity of the process.

Each process may be in any one of the following states -

- New - The process is being created.
- Running - In this state the instructions are being executed.
- Waiting - The process is in waiting state until an event occur like I/O operation completion or receiving a signal.
- Ready - The process is waiting to be assigned to a processor.
- Terminated - the process has finished execution.

It is important to know that only one process can be running on any processor at any instant. Many processes may be ready and waiting.

4. Explain FCFS, SJF (preemptive), priority (non-preemptive and Round Robin (preeptive) and determine waiting time, throughput using each algorithm

→ FCFS -

The simplest CPU scheduling algorithm

Date

is the first-come, first served (FCFS) scheduling algorithm. In this process that requests the CPU first is allocated the CPU first. FIFO (first in first out) queue is used to implement FCFS scheduling.

SJFS - SJFS (short job first scheduling.

shortest-job firs

another scheduling algorithm. In this algorithm as soon as the CPU is available, it is assigned to the process that has the smallest next CPU burst. If two processes have been the some length next CPU burst, FCFS scheduling is used to server the process which come first in the queue.

Priority -

A priority (scheduling) is associated with each process, and the CPU is allocated to the process with the highest priority, Equal-priority processes are scheduled in FCFS order.

An SJFS algorithm is simply a priority algorithm where the priority (p) is the ~~universe~~ inverse of the (predicted) next CPU burst. The larger the CPU burst, the lower the priority and vice versa.

Round-Robin-scheduling-

- The round-robin (RR) scheduling algorithm is designed for timesharing system It is similar to FCFS scheduling, but preemption is added to switch between processes.
- A small unit of time, called a time quantum (or time slice), is defined. A time quantum is generally from 10

Date

to 100 milliseconds.

Algorithm:

1. FCFS

1. Input the processes along with their burst time (bt).
2. Find waiting time (wt) for all process
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 →
5. Find turnaround time = waiting-time + burst-time for all processes.
6. Find average waiting time = total- waiting time'
7. Similarly, find average turnaroundtime = total- turn-around- time

2 SJFS -

1. Traverse until all process gets completely executed.
a) Find process with minimum remaining time at every single timelop.
b) Reduce its time by 1.
c) Check if its remaining time becomes 0.
d) Increment the counter of process completion.
e) completion time of current process = current-time + 1
f) calculate waiting time for each completed process wt[i] = completion time - arrival-time - burst-time
g) Increment time lop byone.

2. Find turnaround time (waiting-time + burst-time)

Sinhgad Institutes | Date

3. Priority-

1- first input the processes with their burst time and priority.

2- sort the processes, burst time and priority

3- Now simply apply FCFS algorithm.

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[].

2- create another array wt[] to store waiting time of processes. Intialize 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 is not doneyet

a- If rem-bt[i] > quantom

i) t = t + quantom

ii) bt-rem[i] = quantom;

c- Else

i) t = t + bt rem[i];

ii) wt[i] = t - bt[i]

iii) bt-rem[i] = 0;

Design Diagram: Class Diagram
use case diagram
sequence Diagram

Input: 1 Enter the number of processes
2. Enter burst time and arrival time of each process.

Sinhgad Institutes

Date

Test case: 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

Conclusion: CPU policies implemented successfully.

Sinhgad Institutes | Date

Assignment 5

Problem statement: Write a JAVA Program (using oop features) to implement paging simulation using

1. FIFO
2. least Recently Used (LRU)
3. optimal algorithm.

Objectives:

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

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 the required new page is called as page replacement. And the techniques used for it are called as page replacement algorithm

NEED OF PAGE REPLACEMENT:

Page replacement is use primarily for the virtual memory management because in virtual memory paging system princip

Date

issue is replacement i.e which page is to be removed so as to begin bring in the new page, thus the use of the page replacement algorithm. Demond pagin is the technique used to increase system throughput. To implement demond paging page replacement is primary requirments If a system has better page replacement technique it improves demond paging which in terns drastically yields system performance agains.

PAGE REPLACEMENT POLICIES:

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

Page Replacement Algorithm:

1. FIFO:

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

Sinhgad Institutes | Date

2. Optimal Page Replacement Algorithm :

Replace the page that will not be used for longest period of times as compared to the pages in main memory. An optimal page replacement algorithm has lowest page fault rate of all algorithm. It is called as OPT or MIN.

Advantage:

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

Disadvantages.

- 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 page's last usage. It uses the recent past as an approximation of the near future, then we can replace the page that has not been used for the longest period of the time i.e the page having larger idle time is replaced.

Advantage-

- The LRU policy is often used for page replacement and is considered to be good.

Sinhgad Institutes | Date

Disadvantages :

1) It is very difficult to implement
2) Require substantial hardware assistance.
3) The problematic determination of the order for the order for the frames defined by the time of last usage.

Algorithm :-

FIFO -

1. Start the process
2. Read number of pages n
3. Read number of pages no
4. Read page numbers into ~~one~~ an array a[i]
5. Initialize avail[i]=0 to check page hit
6. Replace the page with circular queue, while replacing check page availibility in the frome place avail[i]=1 if page is placed in the frome count page fault
7. print the results.
8. Stop the process.

2. ~~Lis~~ Least Recently used -

1. Start the process
2. Declare the size
3. Get the number of pages to be inserted.
4. Get the value.
5. Declare counter and stack

Sinhgad Institutes | Date

6. select the least recently used page by countervalue.
7. stack them according the selection.
8. Display the values.
9. Stop the process.

Design
optimal

1. Start program
2. Read Number of pages and frames.
3. Read each page value.
4. Search for page in the frames.
5. If not available allocate free frome.
6. If NO frames is free replace the page with the page that is least used.
7. Print page Number of page faults.
8. Stop processes.

Design Diagrom: 1. class Diagrom

Input : 1. No. of frames
2. No. of pages
3. page sequence.

output: 1. sequence of allocation of pages in frames
2. cache hit and cache missratio

Test cases:- 1. Test the page hit and miss ratio for different size of pageframes.

Sinhgad Institutes | Date

2. Test the page hit and miss ratio for both algorithm with different page sequence.

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement: Not specific

Conclusion: successfully implemented all page replacement policies.