Daffodil International University

ASSIGNMENT: FALL 2019

COURSE CODE: CIS 323

OPERATING SYSTEMS

Submitted To

MD. AKTARUZZAMAN PRAMANIK

LECTURER

DEPARTMENT OF CIS

DAFFODIL INTERNATIONAL UNIVERSITY

Submitted By

MD. IBRAHIM KHALIL

ID: 182-16-326

DEPARTMENT OF CIS

DAFFODIL INTERNATIONAL UNIVERSITY

Submission Date: 14.12.2019

Table of Contents:

Introduction:

This document is about my academic assignment of **"Operating Systems"** subject. In this document I have discuss about different operating system design its mechanism. I have also discussed about operating systems best performance, efficiency etc. Also I have given the screenshot of provided problem – I have made the solution and finally in task 2 I have described the challenges I have faced while coding. This document is all about it.

Task – 1

Part 1 – Introduction:

I have chosen **"Memory Management"** operating system design. Memory management is the functionality of an operating system which handles or manages primary memory and moves processes back and forth between main memory and disk during execution. Memory management keeps track of each and every memory location, regardless of either it is allocated to some process or it is free. It checks how much memory is to be allocated to processes. (Tutorials Point, 2019)

Purpose in OS area:

The purpose of Memory Management is the set of logical addresses that a process references in its code. Here the three types of addresses used:

Serial No.	Memory Management	Description
01	Symbolic Addresses	The addresses used in a source code. The variable names, constraints, and instruction labels are the basic elements of the symbolic address space.
02	Relative Address	At the time of compilation, a compiler converts symbolic address into relative's address.
03	Physical Address	The loader generates these addresses at the time when a program is loaded into main memory.

What type Mechanism Used to Fulfill the Purpose:

There are lot of mechanism is used for handling memory management is operating system. Some of them are:

- **Memory Partition:** This is mechanisms where process and data located by making partition on memory, like – single-partition or multiple-partition.
- **Fragmentation:** As processes are loaded and removed from memory, the free memory space is broken into little pieces by two processes, like – external and internal fragmentation.
- **Swapping:** Swapping is mechanisms in which a process can be swapped temporarily out of main memory (or move) to secondary storage (disk). The system swaps back the process from the secondary storage to main memory.
- **Paging:** Paging is a memory management mechanism in which process address space is broken into blocks of the same size called **pages**. The size of the process is measured in the number of pages. (Le, 2017), (Wikipedia , n.d.)

Part 2 – System Examples:

Description of each Operating System: (Here I have provided three (3) operating systems description and describe how it realizes the concept).

I. Android OS:

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touch screen mobile devices such as smart phones and tablets.

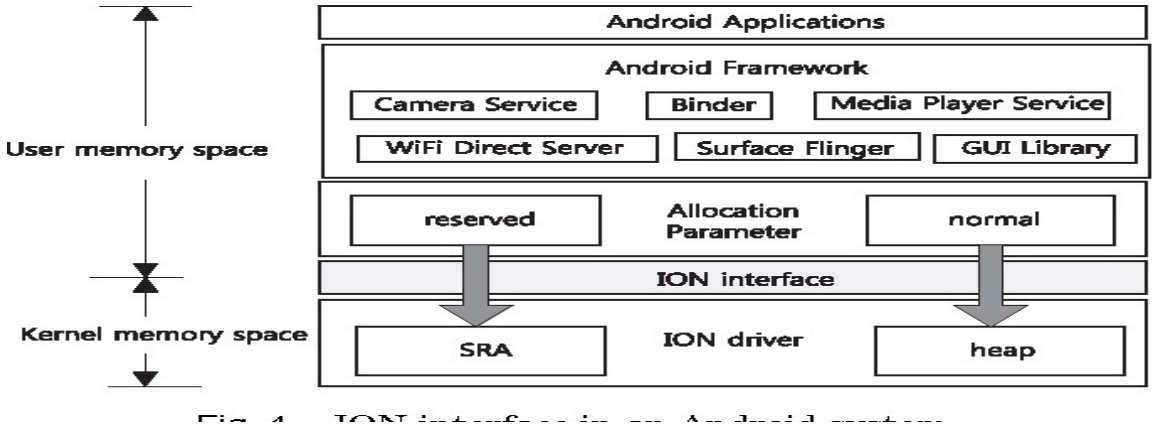

Figure 1: Android OS Memory Management. (Andriod, n.d.)

Concept of Android for Memory Management:

- Android shares the same dynamic RAM across processes using explicitly allocated shared memory regions.
- Most static data is **mmapped** into a process. This technique allows data to be shared between processes, and also allows it to be paged out when needed.

II. Windows OS:

Windows is the name of an operating system named Windows 1.0 in 1985 up to their latest release named Windows 10 in 2015. Windows is built on 2 different bases depending on the version of Windows you're using:

- Windows 1.0, 2.0, 3.1, 3.11, 95, 98, and ME run on MS-DOS
- Windows NT, 2000, XP, Vista, 7, 8, 8.1, 10, and CE run on NT

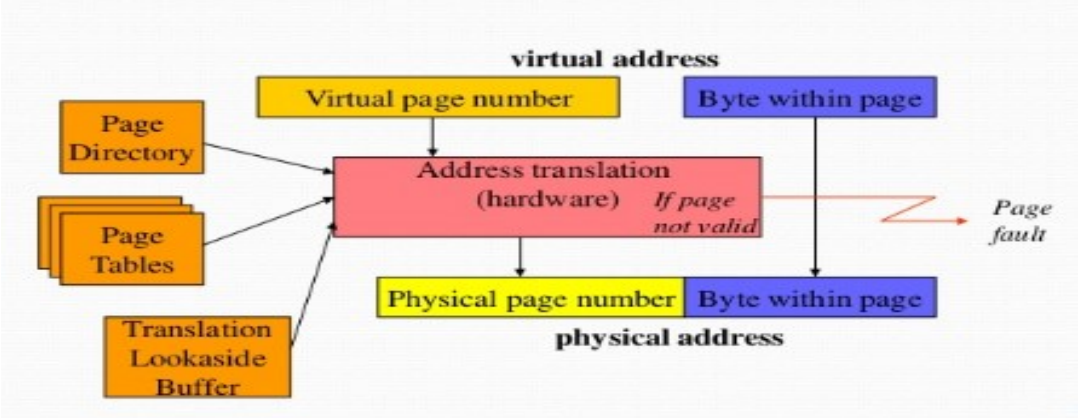

Figure 2: Windows OS Memory Management. (Google, n.d.)

Concept of Windows for Memory Management:

The Windows Memory Manager has two first-in-rank responsibilities:

- The virtual address space might be larger or smaller than the physical memory on the machine.
- The second one is paging some of the contents of memory to disk when it becomes overcommitted. That is currently available-and bringing the contents back into physical memory as needed.

III. MAC OS:

It was originally called “Mac OS X”, then later “OS X”, and is now called simply “Mac OS”. Based on the Open STEP operating system, this is a new OS, and is not compatible with what was later termed “Classic Mac OS” and was not available on Intel-based versions.

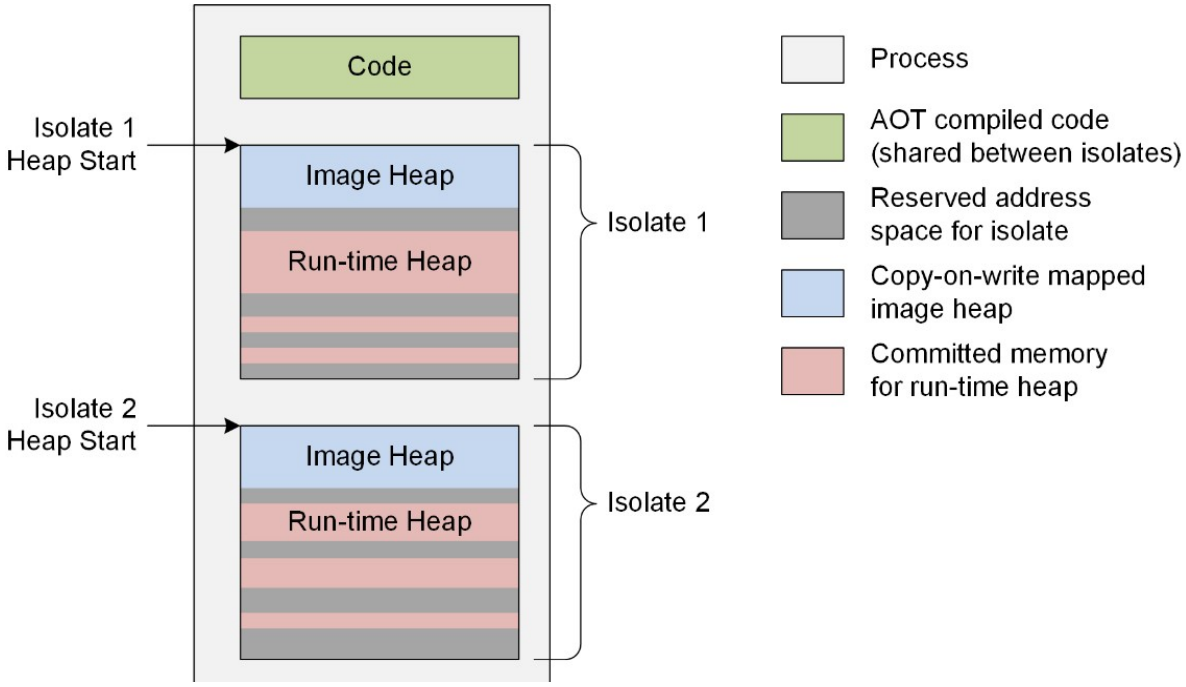

Figure 3: MAC OS Memory Management. (Google, n.d.)

Concept of MAC OS for Memory Management:

- It divides the available RAM into two broad sections. It reserves for itself a zone or makes partition of memory known as the system partition.
- The system partition always begins at the lowest addressable byte of memory and extends upward. (Tutorials Point, 2019)

Part 3 – Compare & Evaluation:

Critical Comparisons and philosophies on Three (3) actual systems in OS based on **"Memory Management"** OS Design:

Serial No.	Basic For	Memory Management and Mechanism	Philosophies
01	**Android**	**1)** The **Android Runtime (ART)** and **Dalvik Virtual Machine** use paging and **memory-mapping (mmapping)** to manage memory. **2)** It uses flash memory for persistent storage like hard drives in a desktop.	**1)** Android had always been focused on giving the users raw power and performance, and overlooked the aspect that a personal. **2)** Android was carved out from the heart of Linux.
02	**Windows OS**	**1)** The virtual memory system starts to thrash and performance is greatly degraded. **2)** Low memory and CPU usage. Consist of Kernel memory system starts to trash and perform is greatly degraded. The programs will run, but very slowly.	**1)** Windows fully supports symmetric multi processing with multi-user support. **2)** Windows is NOT limited to the GUI. As time has gone on, each new release has added the ability to perform more operations via text/script commands.
03	**MAC OS**	**1)** Fully-integrated virtual memory system. Providing up to 4 gigabytes of addressable space per 32-bit process and approximately 18 Exabyte of addressable space for 64-bit process. **2)** The virtual memory system uses hard disk storage (SWAP) to hold data not currently in use.	**1)** MAC OS energies as a business are very much focused on the creative fields, which is one of its biggest markets. **2)** MAC OS invented Quick time, a multimedia format which allowed the user to play all standard audio and video formats seamlessly.

(Teresi, 2018), (Logic Child, 2008), (Chowdhury, 2015)

Recommendation based on Performance/Efficiency:

I've tried to be fair and objective – to find out the best OS from these based on performance or efficiency. In my opinion the **MAC OS is better** than other two Oss. MAC's is more useful for critical work or performance.

Android or Windows offers some great computer hardware options, but the selection by the massive availability of a diverse array of **MAC OS**. As for internal components - things like the CPU, graphics card, and storage - there's no contest. **MAC OS** gives you a lot more way in configuring a system with the components you want, and more flexibility to upgrade later.

Number of the Words for (Task 1): 1053

Task – 2 (Lab)

Subtask – 1

Program's output screenshot (Using C Language):

```
                                  ------------------------------------------------------
                                       # OPERATING SYSTEMS LAB ASSIGNMENT #
                                  ------------------------------------------------------
                                           MADE BY - MD. IBRAHIM KHALIL
                                                ID-> 182-16-326
                                  ------------------------------------------------------



-------------------------------------------------------
 Choose Any One:
-------------------------------------------------------
         0. To Give Input:
         1. Show (FCFS) Gantt Chart:
         2. Show (SJF Preemptive) Gantt Chart:
         3. Show (Round Robin) Gantt Chart:
         4. Show Comparison Table:
         5. Exit
-------------------------------------------------------

 Enter Your Choice: 0

 Enter Total No. of Processes : 4

 PROCESS [1] Enter Process Name : p1
 Enter Arrival Time : 0
 Enter Burst Time : 14

 PROCESS [2] Enter Process Name : p2
 Enter Arrival Time : 3
 Enter Burst Time : 7

 PROCESS [3] Enter Process Name : p3
 Enter Arrival Time : 5
 Enter Burst Time : 11

 PROCESS [4] Enter Process Name : p4
 Enter Arrival Time : 6
 Enter Burst Time : 5
```

Figure 4: Process of Data Input.

```
-------------------------------------------------------
 Choose Any One:
-------------------------------------------------------
         0. To Give Input:
         1. Show (FCFS) Gantt Chart:
         2. Show (SJF Preemptive) Gantt Chart:
         3. Show (Round Robin) Gantt Chart:
         4. Show Comparison Table:
         5. Exit
-------------------------------------------------------

 Enter Your Choice: 1


PROC.    A.T.     B.T.     W.T      T.A.T
---------------------------------------

 p1       0        14       0        14
 p2       3        7        11       18
 p3       5        11       16       27
 p4       6        5        26       31
---------------------------------------


                                GANTT CHART (FCFS):
                                ****************************************
                                0| p1 |14| p2 |21| p3 |32| p4 |37
```

Figure 5: FCFS Gantt chart.

```
Enter Your Choice: 2


                        GANTT CHART (SJF_Preemptive):
                        ****************************************
                        0| p1 |3| p2 |10| p4 |15| p1 |26| p3 |37
```

Figure 6: SJF Preemptive Gantt chart.

```
Enter Your Choice: 3

Enter Time Slice for Round Robin: 4


                    GANTT CHART (Round-Robin):
                    ************************************
                    0| p1 |4| p2 |8| p3 |12| p4 |16| p1 |20| p2 |23| p3 |27| p4 |28| p1 |32| p3 |35| p1 |37
```

Figure 7: Round Robin Gantt chart.

```
--------------------------------------------------
 Choose Any One:
--------------------------------------------------
        0. To Give Input:
        1. Show (FCFS) Gantt Chart:
        2. Show (SJF Preemptive) Gantt Chart:
        3. Show (Round Robin) Gantt Chart:
        4. Show Comparison Table:
        5. Exit
--------------------------------------------------

 Enter Your Choice: 4

 Enter Time Slice for Round Robin: 4


                    ************************************************************************
                                          # COMPARISON TABLE #
                    ************************************************************************
                     || ALGORITHM    ||  AVG. WAITING TIME   ||  AVG. TRUN-AROUND TIME ||
                    ************************************************************************
                        1. FCFS               (13.25)                (22.50)
                        2. SJF_P.E.           (9.25)                (18.50)
                        3. R. ROBIN           (18.00)                (27.25)
                    ************************************************************************
```

Figure 8: Comparison Table.

Subtask – 2

Description of what challenges I faced while coding:

Serial No.	Challenges	Description
01	Take same input for all three (3) programs.	For taking input I have declare an array and take the arrival time and burst time after taking no of process of the program. Then make and function and this function will pass these data for making calculation of each process.
02	Making Gantt Chart.	Following each algorithm – implement it in code and by sorting them create Gantt chart of each schedule.
03	Different function for each algorithm.	FCFS, SJF, Round-Robin are different algorithm and logic. So, I have created different function for these and call them individually – then the program shows the output.
04	Round Robin problem and its time slice.	For round robin algorithm calculation, I have take another input this is called time slice and calculation are make depend on it.
05	Comparison Table using input data.	This is the most complex part of this program. I have made a different function for making comparison table. All calculation is passed by this function and final output will shown using by this function.

Conclusion:

This work was interesting. I have use and analyze world's best operating system and it's each feature and compares each. Then finally make a report about different kind of operating system its mechanism, how it actually uses in different types of mechanism. And solve the algorithm for lab assignment. I would like to thanks to my course leader **"Aktaruzzaman Pramanik"** sir for adding this types of assignment in our course. I hope this knowledge will help me in my future computing career.

References:

Chowdhury, R. (2015) *One Xtrapixel*, [Online], Available: HYPERLINK "https://onextrapixel.com/ios-vs-android-evolution-of-design-philosophy/" https://onextrapixel.com/ios-vs-android-evolution-of-design-philosophy/ [2019].

Developer (2019), [Online], Available: HYPERLINK "https://www.developer-tech.com/news/2013/may/02/the-pros-cons-of-the-major-mobile-operating-systems/" https://www.developer-tech.com/news/2013/may/02/the-pros-cons-of-the-major-mobile-operating-systems/ [2019].

Geek For Geeks (2019), [Online], Available: HYPERLINK "https://www.geeksforgeeks.org/types-of-operating-systems/" https://www.geeksforgeeks.org/types-of-operating-systems/ [2019].

Le, J. (2017) *Medium*, 28 December, [Online], Available: HYPERLINK "https://medium.com/cracking-the-data-science-interview/the-10-operating-system-concepts-software-developers-need-to-remember-480d0734d710" https://medium.com/cracking-the-data-science-interview/the-10-operating-system-concepts-software-developers-need-to-remember-480d0734d710 [23 November 2019].

Logic Child (2008), [Online], Available: HYPERLINK "https://www.codeproject.com/Articles/29449/Windows-Memory-Management" https://www.codeproject.com/Articles/29449/Windows-Memory-Management [2019].

Teresi, S. (2018) *Teresi.US*, [Online], Available: HYPERLINK "http://teresi.us/html/writing/mac_vs_windows.html" http://teresi.us/html/writing/mac_vs_windows.html [2019].

Tutorials Point (2019), [Online], Available: HYPERLINK "https://www.tutorialspoint.com/operating-system-design-and-implementation" https://www.tutorialspoint.com/operating-system-design-and-implementation [23 November 2019].

Wikipedia , [Online], Available: HYPERLINK "https://en.wikipedia.org/wiki/Comparison_of_operating_systems" https://en.wikipedia.org/wiki/Comparison_of_operating_systems .