SECTION C: EXAM PROJECT 3 REPORT

A P Barnard - 28430093

TABLE OF CONTENTS

INTRODUCTION

Background

Each member of the group was tasked with using their skills in Operating System, specifically, File Systems, Scheduling and Multiple Queues, and Virtual Memory Systems. We had to implement these solutions on Linux and Windows Server Virtual Machines (VM).

What are virtual machines? Discussed in a previous report and analysed below:

Virtual Memory is a concept that was first created in 1961 by J. Fotheringham (Fotheringham, 1961). The idea behind this concept is that every program on a computer has its own address space and that is broken up into smaller chunks called pages. The reason for the chunking of pages is that larger programs are basically too large for the computer to store and process and execute.

Pages are mapped onto physical memory spaces, since pages are made up of contiguous address ranges.

Most virtual memory systems use the concept of Paging. Every program on a computer references a set of memory addresses that executes instructions by copying contents of the memory address that has been indexed using a base and segment register. These program-generated addresses are called virtual addresses and are part of the virtual address space. Virtual addresses are maintained and created by the MMU (Memory Management Unit) (Tanenbaum & Bos, 2015)

Problem Statement

For Section C, I had to implement the concepts of Virtual Memory and Paging on either a Linux or Windows VM.

TASK:

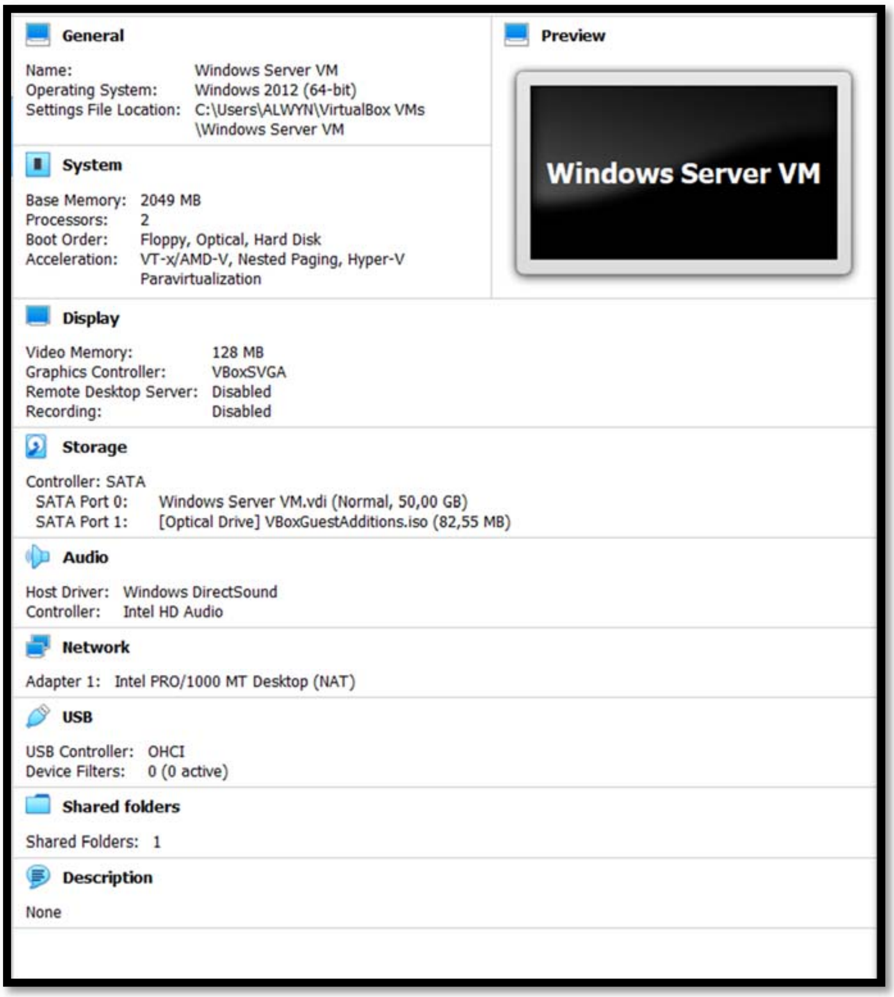
1. Retrieve the amount of memory available on the server.
2. Ask the user for the amount of memory that has to be reserved for the Operating System (OS).
3. Ask the user for the amount of memory that has to be assigned to each Page Frame.
4. Calculate the number of Page Frames that can be created.
5. Simulate Paging by randomly assigning unique page numbers into the page frames.
6. Ask the user for a page to search for.
7. Show if the page is in Memory, or if it causes a page fault.
8. Use a page replacement algorithm to load the page not found into a frame.
9. Deploy the solution on your Virtual Machine.
10. Construct a report on the Project.

CONTENT

The following Virtual Machine was created:

- VirtualBox was used to create the VM

- Windows Server 2012 Virtual Machine with IIS
- Base Memory of 2048 MB
- 2 Processors Used with 100% Execution cap
- Acceleration: VT-x/AMD-V and Nested paging enabled
- 128 MB Video Memory with 2D and 3D Video Acceleration
- Shared folder between Host OS and Guest OS



General

```
Name:                   Windows Server VM
Operating System:       Windows 2012 (64-bit)
Settings File Location: C:\Users\ALWYN\VirtualBox VMs
                        \Windows Server VM
```

Preview

System

```
Base Memory:  2049 MB
Processors:   2
Boot Order:   Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, Hyper-V
              Paravirtualization
```

Display

```
Video Memory:          128 MB
Graphics Controller:   VBoxSVGA
Remote Desktop Server: Disabled
Recording:             Disabled
```

Storage

```
Controller: SATA
  SATA Port 0:   Windows Server VM.vdi (Normal, 50,00 GB)
  SATA Port 1:   [Optical Drive] VBoxGuestAdditions.iso (82,55 MB)
```

Audio

```
Host Driver:  Windows DirectSound
Controller:   Intel HD Audio
```

Network

```
Adapter 1:  Intel PRO/1000 MT Desktop (NAT)
```

USB

```
USB Controller:  OHCI
Device Filters:  0 (0 active)
```

Shared folders

```
Shared Folders:  1
```

Description

None

PROGRAMMING LANGUAGE:

- Visual Studio 2019
- ASP.NET Web Forms Application with additional Web Forms

INPUT COMPONENTS:

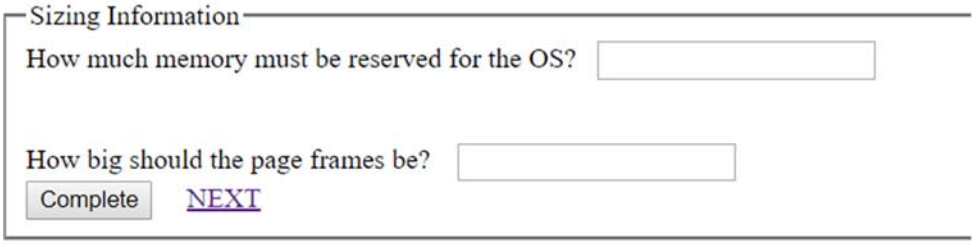
- Textboxes, Buttons and Labels

OUTPUT COMPONENTS:

- Textboxes, Labels, Panels and Buttons
- Command Line results

Forms:

MAINFORM

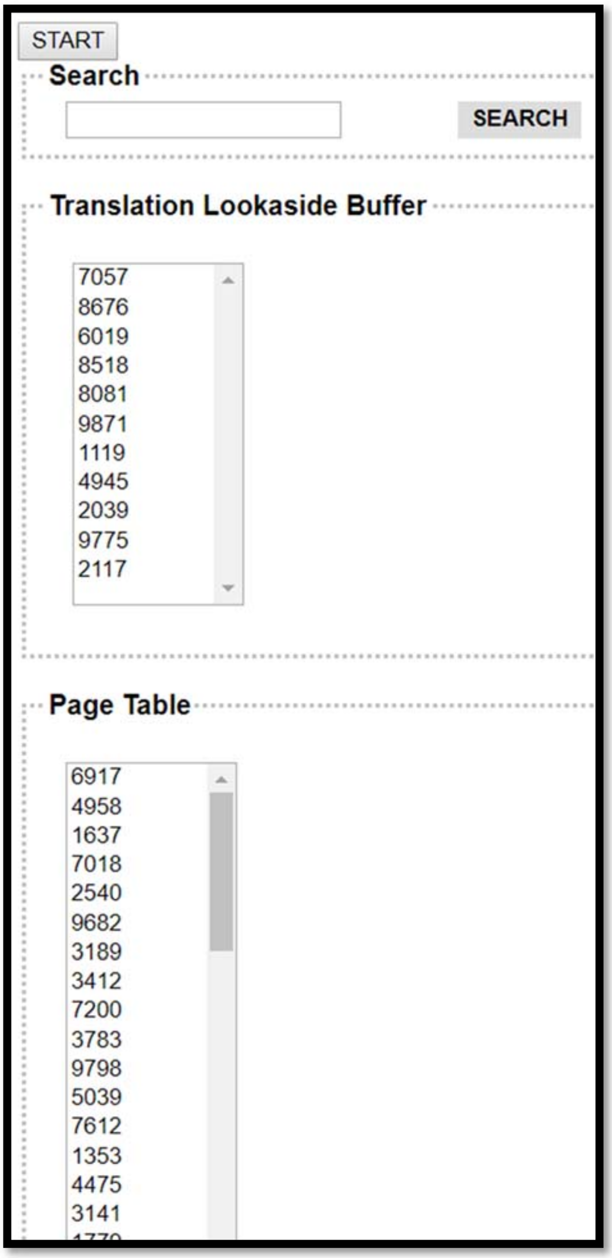


Sizing Information

How much memory must be reserved for the OS?

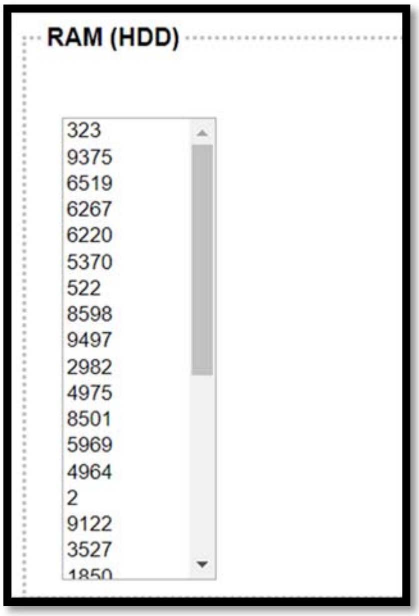
How big should the page frames be?

PAGING FORM



```
7057
8676
6019
8518
8081
9871
1119
4945
2039
9775
2117
```

```
6917
4958
1637
7018
2540
9682
3189
3412
7200
3783
9798
5039
7612
1353
4475
3141
1770
```



```
323
9375
6519
6267
6220
5370
522
8598
9497
2982
4975
8501
5969
4964
2
9122
3527
1850
```

RESULTS

The results that can be seen on the above forms are as follows. If incorrect input is received the Validators will show that there are incorrect values. Otherwise the program will look for a value that is either in the TLB, Page Table or in RAM. If no such value is found, then an error will be shown that it does not exist in memory.

Every possible precaution was taken to ensure data validity and where invalid data was entered the user will be prompted to change those values.

CONCLUSION

Virtual memory plays an integral role in Operating Systems. It is the foundation that was used to build virtual address spaces and the accompanying concepts. With it we would not be able to run multiple Operating Systems on one computer. Limitations to virtual memory are “virtually” impossible.

REFERENCES

Fotheringham, J.: ‘‘Dynamic Storage Allocation in the Atlas Including an Automatic Use of a Backing Store,’’ Commun. of the ACM, vol. 4, pp. 435–436, Oct. 1961.

Tanenbaum, A. S. & Bos, H. 2015. Modern operating systems. 4th ed. Upper Saddle River, NJ: Pearson Prentice Hall.

.