BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING
Department: Software Engineering
Year: 2nd Year, 2017 E.C

NAME ID

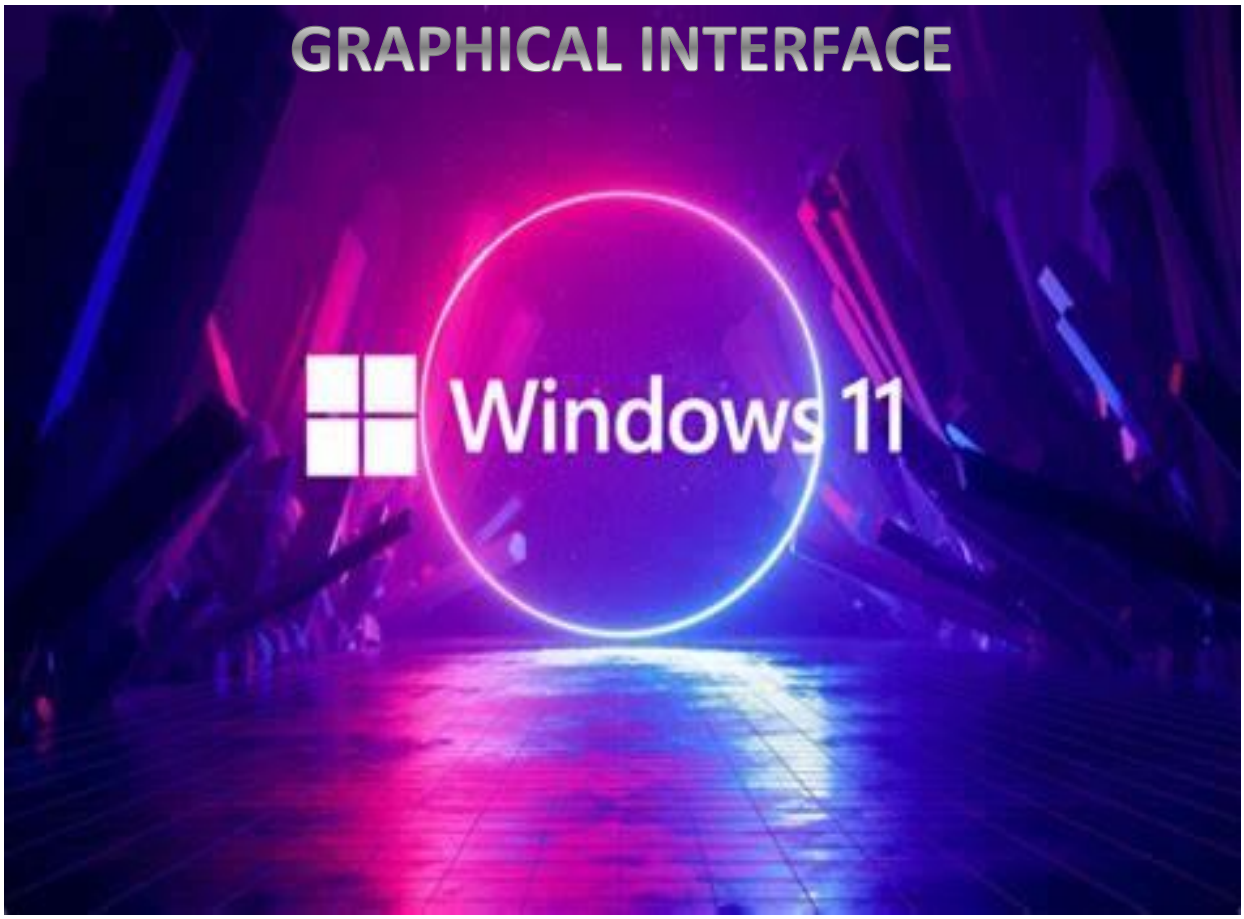
GATWECH DENG KUN……1510065

SUBMITTED TO: Mr. WONDIMU BAYE

Submission Date: 16/07/2017 E.C.

OPERATING SYSTEM ASSIGNED: WINDOWS 11

VIRTUALIZATION TOOL: VMWARE WORKSTATION



WORLD-VIEW

PROJECT-WORK

OPERATING SYSTEM AND SYSTEM PROGRAMMING

1. Installation of operating system in virtual environment tools (VMware workstation, oracle vm virtual box, and etc...). Installation may not necessary if the technology tool is outdated, has no long term support (LTS), and no hardware support only. Which OS you are going to install refer on the attached document. Your name list as well as OS list is incorporated. Contents to cover when you prepare documentation:

a. Introduction (background, motivation)

b. Objectives

c. Requirements i. Hardware ii. Software

d. Installation steps: i. Include snipped images ii. When you create an account, you should name by your full-name

e. Issues (problem faced): when you face problems, list or snip the problem/s.

f. Solution: i. if you have the solution for the problem listed in above (e), list the solution

g. File system support.

h. Which file system support(NTFS, FAT32, exFAT, ext4, Btrfs, ZFS, HFS+, APFS) and why

i. Advantage and disadvantage.

j. Conclusion. j. Future outlook /recommendation.

2. Briefly explain the what, why, and how virtualization in modern operating system.

4. Implement system calls.

4. IMPLEMENT SYSTEM CALLS.

1. System call implementation is the process of accessing computer hardware.

2. Is the fundamental mechanism by which user-space programs request services from the operating system kernel.

3. It allows processes to request privileged operation like:

- File I/O
- Process control
- Device management
- Memory allocation
- Inter-process communication

MLOCK () system call is a POSIX function that locks memory pages into RAM, preventing them from being paged to disk.

But windows has different APIs for similar functionality.

Windows doesn't have mloc () directly, but provides

1 virtualLock() –lock pages

2 virtualUnlock() –unlock memory pages

- Here is my code of locking and unlocking the memory of the disk, am using c++ language.

```
#include <windows.h>

#include <iostream>

int main() {

   // Allocate memory

   const size_t size = 4096; // 4KB

   void *ptr = VirtualAlloc(NULL, size, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);

if (!ptr) {

      std::cerr << "Failed to allocate memory!" << std::endl;

      return 1;

   }   // Lock the memory to prevent paging

   if (VirtualLock(ptr, size)) {

      std::cout << "Memory locked successfully!" << std::endl;

   } else {

      std::cerr << "Memory lock failed!" << std::endl;

   }

// Simulate work

   Sleep(5000);

// Unlock the memory

   VirtualUnlock(ptr, size);

   VirtualFree(ptr, 0, MEM_RELEASE);

   std::cout << "Memory unlocked and freed." << std::endl;

return 0;

}
```

Here I have to type the Desktop directory cd Desktop

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP> cd Desktop
PS C:\Users\HP\Desktop>
```

I have to type notepad for writing my code in the editor

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP> cd Desktop
PS C:\Users\HP\Desktop> notepad mlock.cpp
```

Here I have written the code and I will press ctrl+s for saving

```
#include <windows.h>
#include <iostream>

int main() {
    // Allocate memory
    const size_t size = 4096; // 4KB
    void *ptr = VirtualAlloc(NULL, size, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);

    if (!ptr) {
        std::cerr << "Failed to allocate memory!" << std::endl;
        return 1;
    }

    // Lock the memory to prevent paging
    if (VirtualLock(ptr, size)) {
        std::cout << "Memory locked successfully!" << std::endl;
    } else {
        std::cerr << "Memory lock failed!" << std::endl;
    }

    // Simulate work
    Sleep(5000);

    // Unlock the memory
    VirtualUnlock(ptr, size);
    VirtualFree(ptr, 0, MEM_RELEASE);
    std::cout << "Memory unlocked and freed." << std::endl;

    return 0;
}
```

Now I have to compile it using g++ mlock.cpp –o molck and run it using ./ mlock command

```
PS C:\Users\HP\Desktop> g++ mlock.cpp -o mlock
PS C:\Users\HP\Desktop> ./mlock
Memory locked successfully!
Memory unlocked and freed.
PS C:\Users\HP\Desktop>
```