**MINI OPERATING SYSTEM**

**Project Report**



**Fall 2022**

**CSE-302L Systems Programming Lab**

Submitted by: **Ebtihaj Abdullah Abdur-Rahman**

Registration No. : **20PWCSE1885, 20PWCSE1878**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Engr. Abdullah Hamid**

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**ABSTRACT:**

Operating System is an essential part of every computer system. It acts as a communication bridge between hardware and the user. OS is a very big piece of code but here we have limited that to a small bit of custom designed system calls and function as well as libraries hence the name MINI OS.

**TOOLS USED:**

We have used the following software tools to develop this OS.

* GCC Compiler : To compile C code.
* NASM : To compile Assembly code.
* QEMU : To build terminal GUI in absence of a virtual machine client.
* GRUB : To make an iso image of our OS.
* Linker: To link object and assembly files into a binary (.bin) file.
* VMWARE : To build a virtual machine host and test our iso file.

**LIBRARIES USED:**

We have written the libraries code ourselves hence the libraries are all custom built.

**SYSTEM CALLS USED:**

We have written the system calls code ourselves hence the system calls are all custom built.

**CODE FLOWCHART:**

kasm.o

PROJECT

Kc.o

link.ld

kernel.asm

kernel.c

build.sh

grub.conf

kernel.bin

grub

boot

system.o

screen.o

kb.o

string.o

system.c

system.h

string.c

string.h

types.h

Screen.h

screen.c

kb.h

kb.c

boot\_folder

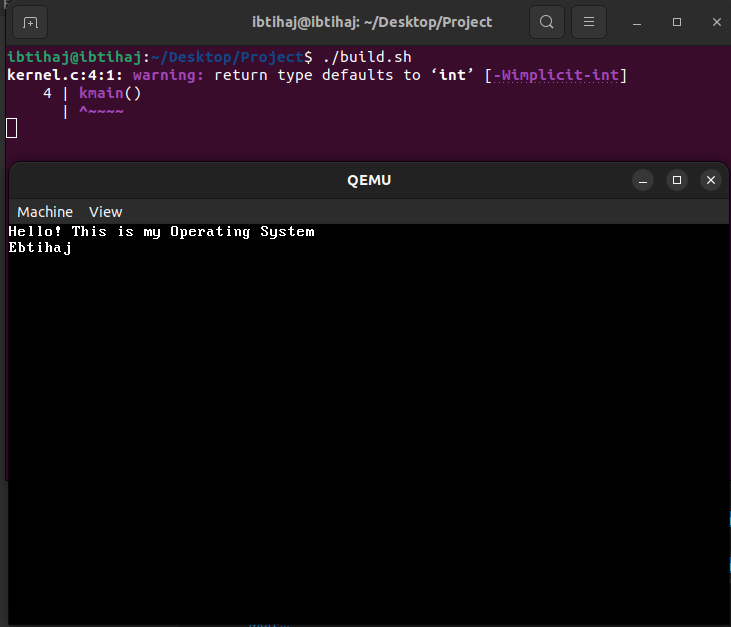
obj

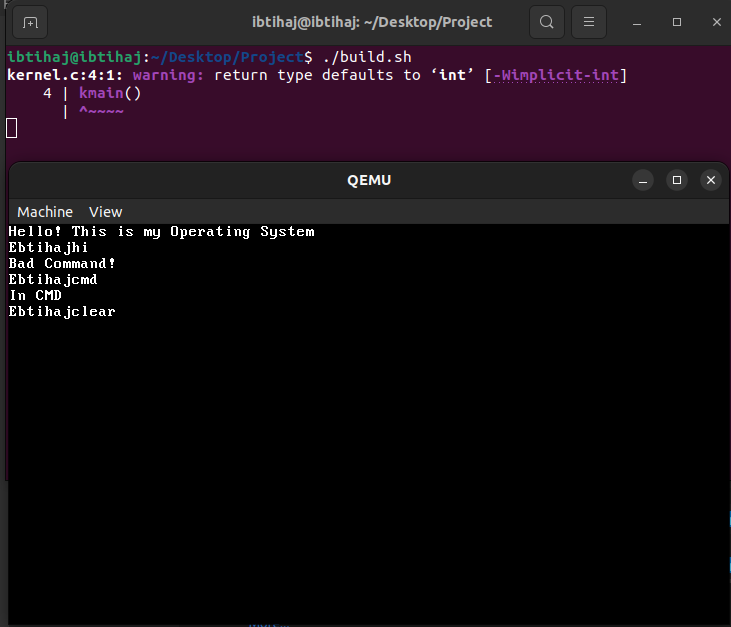
include

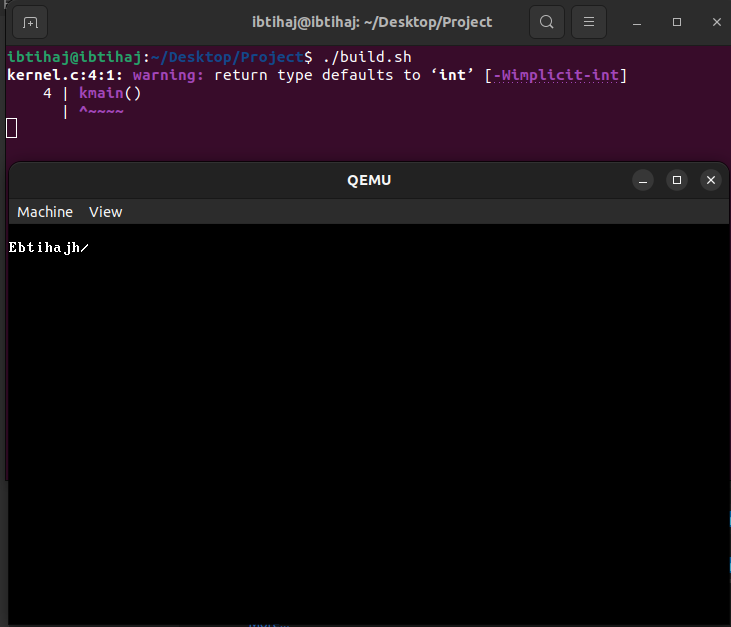
**CODE WORKING:**

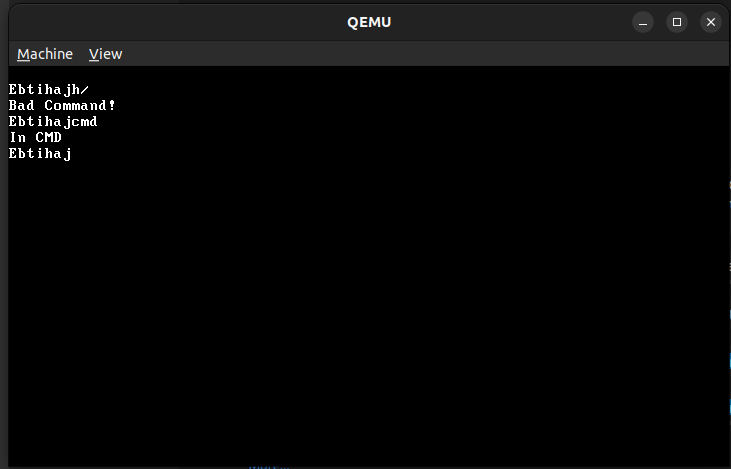
We have main execution lines for examples gcc nasm and grub commands in build.sh. kernel.c contains c code for kernel and the main global function kmain. Kernel.asm is the assembly code for booting the OS. Link.ld contains linker code. Kc.o and kasm.o are the object files. In include folder, we have libraries (header files and their source files). In obj folder, we have the object files of the source files which are linked with kasm to make binary file. In boot\_folder, we have the binary file for kernel and in grub folder, we have config file of grub which are the settings to make an iso.

**OUTPUT:**

****

****

****

****