**CS2006 OPERATING SYSTEMS**



**PROJECT REPORT**

**LIGHTWEIGHT VIRTUAL MACHINE**

**COURSE INSTRUCTORS:**

Dr.Ghufran Amed

Miss Anaum Hamid

**MEMBERS:**

* 21K-3429 ASHHAD ABID
* 21K-3313 MIRZA UMAIR BAIG

**OBJECTIVE:**

The main purpose of a Lightweight Virtual machine is to operate multiple operating systems at the same time, from the same piece of hardware.

**INTRODUCTION:**

This project is an attempt to make a Virtual Machine that takes assembly language code and runs it on a lightweight VM that we have made.

**BACKGROUND:**

A virtual machine (VM) is software that runs programs or applications without being tied to a physical machine. In a VM instance, one or more guest machines can run on a host computer.

Each VM has its own operating system and functions separately from other VMs, even if they are located on the same physical host. VMs generally run on servers, but they can also be run on desktop systems or even embedded platforms. Multiple VMs can share resources from a physical host, including CPU cycles, network bandwidth, and memory. Because the software is separate from the physical host computer, users can run multiple OS instances on a single piece of hardware, saving a company time, management costs, and physical space. Another advantage is that VMs can support legacy apps, reducing or eliminating the need and cost of migrating an older app to an updated or different operating system. In addition, developers use VMs in order to test apps in a safe, sandboxed environment.

**PLATFORM AND LANGUAGES:**

1: Ubuntu Terminal

2: The language used for this project is C.

3: Assembly Language Object Code.

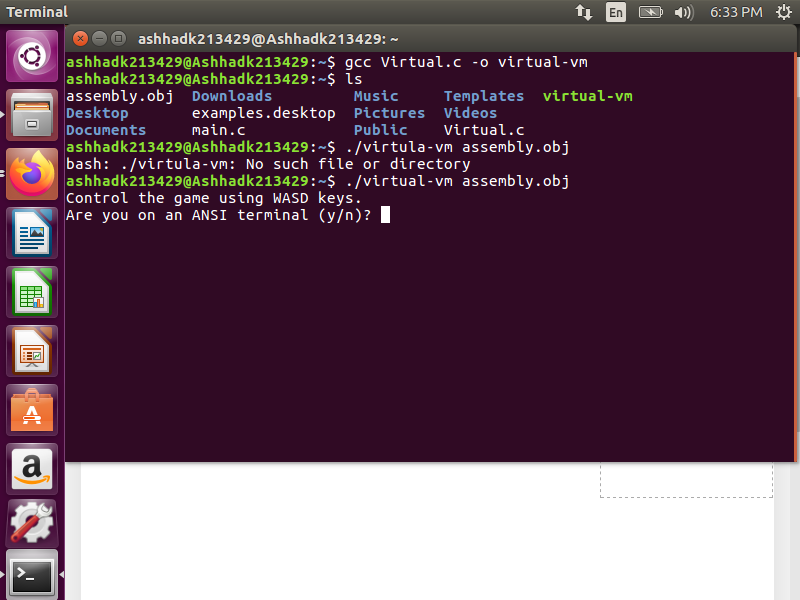
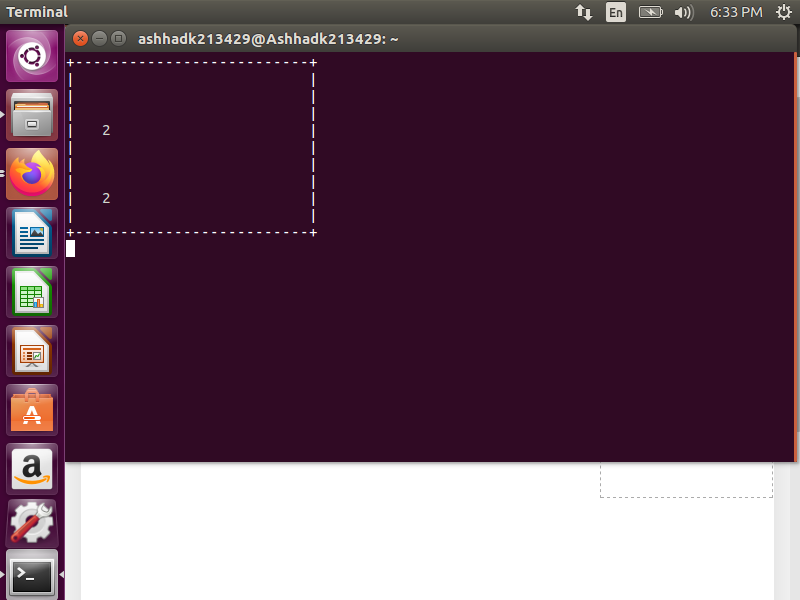
**METHODOLOGY:**

1: Firstly, VM is compiled

2: Then, all opcodes of directives, jump conditions are set in one file.

3: After We can pass any assembly language object file to run It on a Lightweight virtual machine

**RESULTS**



**CONCLUSION:**

The given code presents a virtual machine that simulates the execution of assembly code. It utilizes a specific instruction set architecture and bytecode format to interpret and run assembly programs. The code incorporates essential functionalities such as instruction handling, memory management, arithmetic operations, and conditional branching. By providing a platform to emulate assembly code execution, this virtual machine serves as a valuable tool for educational purposes, experimentation, and the development and testing of assembly-based applications.