**CS 6235 Project Proposal**

**X86 Virtual Machine Implementation**

Group member: Xu Ha, Jia Zhao

1. **Goal**

This project aims at building an x86 virtual machine of IBM-PC/AT architecture, which is a hardware emulator takes the form of a hardware device and can support some old commercial operating systems, such as MS-DOS. We would like to study the principle of the virtual machines, investigate how the devices interact with each other, and try to optimize it based on the knowledge of Real Time Systems.

1. **Overview**

The virtual machine is a software implementation of a computer that can execute programs like a physical machine. In this project, we are going to implement a system virtual machine, which provides a complete system platform and supports the execution of a complete operating system. In another words, it is an efficient and isolated duplicate of a real machine. The resources of this virtual machine are limited; they can be assigned by the users and cannot exceed the physical limits. This virtual machine will be developed in C or C++, and the host operating system is Microsoft Windows. To simplify the problem, it is a 16-bit virtual machine of IBM-PC/AT architecture, and its object guest operating system is MS-DOS 6.22. Several basic devices, such as the floppy drive, keyboard and so on, will be developed for this virtual machine. The manual and documentation of these devices can be collected from the Internet. Building such a system virtual machine will help us gain a deeper understanding in the real computer architecture, and learn how the devices can work efficiently with each other. This virtual machine should be stable and portable. We will also try to optimize the system to make it more efficiency, for example, to make the resource allocation more reasonable.

1. **Architecture**

We want to gain the lowest degree of coupling among the virtual devices, and wish to build a system which is portable, i.e. to make it platform independence. Therefore it has been divided into three parts: supportive module, virtual machine module and platform-related module.

* 1. **Supportive Module**
     1. Debugger. The debugger is used to test the correctness of the virtual CPU and other virtual devices. It is not only the hexadecimal memory editor, but also the assembler, disassembler and instruction executor. We perform most of our test on this debugger while building the virtual machine.
     2. Console. The console provides the user an interface to modify the virtual machine settings before launching it.
  2. **Virtual Machine Module**
     1. Central Processing Unit, Intel 8086
     2. Programmable Interrupt Controller, Intel 8259
     3. Direct Memory Access Controller, Intel 8237
     4. Programmable Internal Timer, Intel 8253
     5. Floppy Drive Controller, Intel 8272
     6. Keyboard Controller, Intel 8042
     7. Video Display Controller, Intel 8275
     8. BIOS. It is a file which contains all interrupt service routines (ISR) and is to be loaded when powering on the virtual machine.
  3. **Platform-related Module**
     1. Keyboard, which accepts input from Window in Win32 Platform
     2. Display, which prints the output on Window in Win32 Platform

1. **Timeline**

|  |  |
| --- | --- |
| Week 1 | Investigation and manual collection |
| Week 2 | Implementation of supportive module |
| Week 3 | Implementation of virtual machine module |
| Week 4 |
| Week 5 |
| Week 6 | Implementation of platform-related module |
| Week 7 | Testing, documentation and optional (optimization) |
| Week 8 |

1. **References**
   1. Wikipedia, <http://en.wikipedia.org/wiki/Virtual_machine>
   2. VMware, <http://www.vmware.com/virtualization/what-is-virtualization.html>
   3. QEMU, <http://wiki.qemu.org/Main_Page>
   4. Microsoft Virtual PC, <http://www.microsoft.com/windows/virtual-pc/>