
VE482 - Lab 1

Introduction to Operating Systems

Kexuan Huang 518370910126

September 16, 2021

I. Hardware overview

1. In the computer

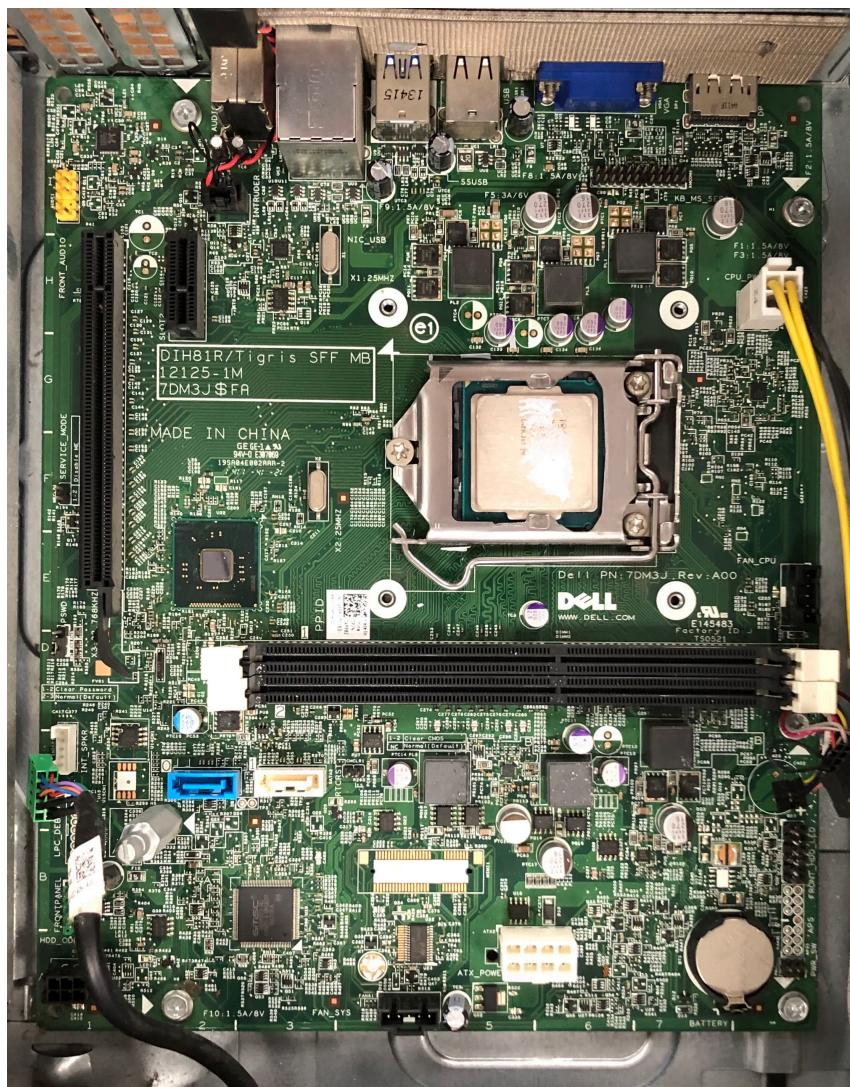


Figure 1: The motherboard

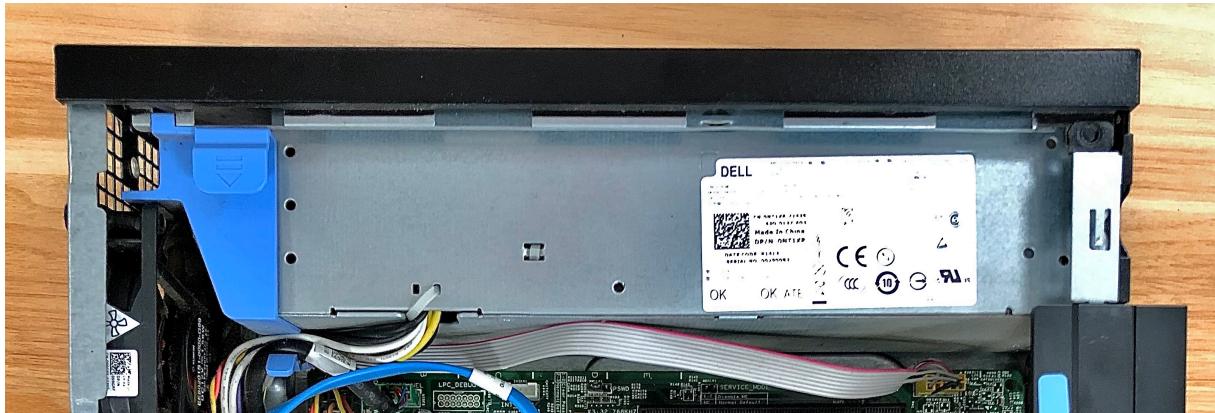


Figure 2: The PC power supply



Figure 3: A Hard Disk Drive



Figure 4: A PCI card



Figure 5: An Optical dist drive

2. On the motherboard



Figure 6: The RAM



Figure 7: The North bridge

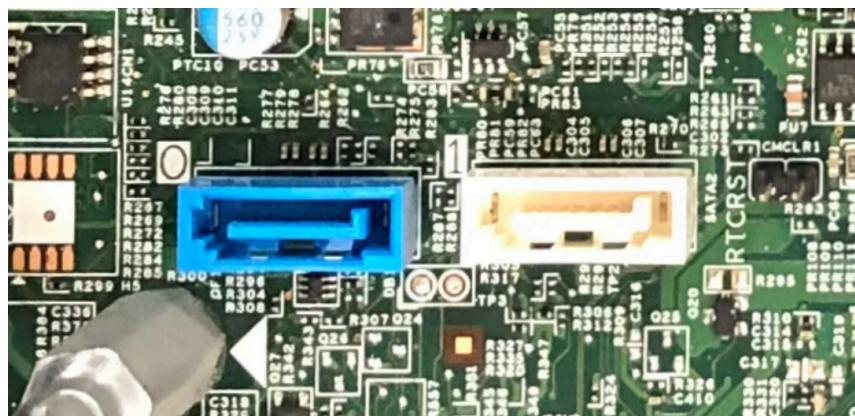


Figure 8: A SATA socket



Figure 9: The CPU



Figure 10: The battery

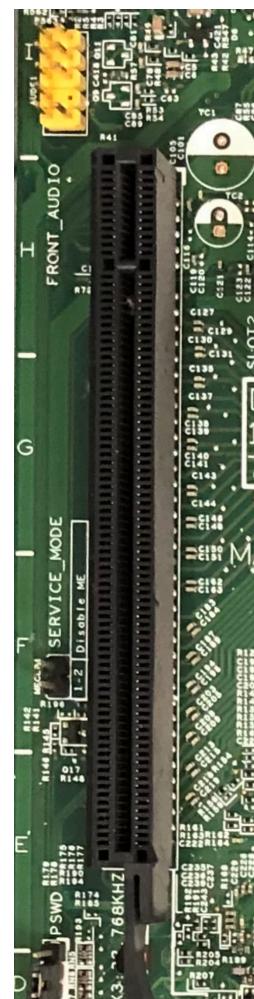


Figure 11: A PCI/PCI-e slot

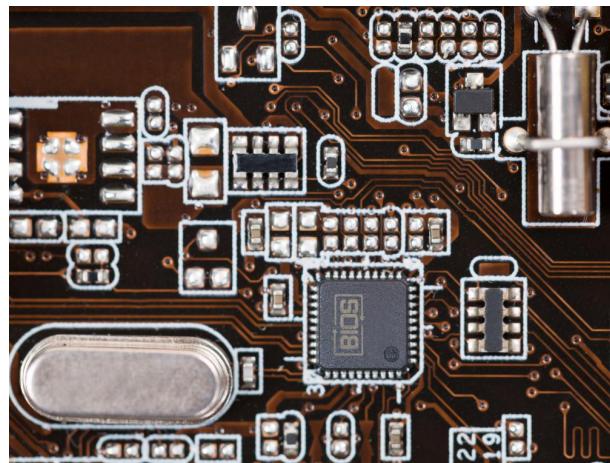


Figure 12: The BIOS

3. Q&A

Where is the CPU hidden, and why?

The CPU is hidden under a fan with a metal cover on it to dissipate the heat generated when working.

What are the North and South bridges?

North bridge is a chip located in the direction towards North in the motherboard whose main function is to manage the communications between the Central Processing Unit and parts of motherboard.

South bridge is a chip located to the South of Peripheral Component Interconnect (PCI) bus in the motherboard, whose main function is to control the IO functioning.

How are the North and South bridges connected together?

They are connected by an Internal Bus.

What is the BIOS?

BIOS is the firmware used to perform hardware initialization during the booting process (power-on startup), and to provide runtime services for operating systems and programs.

Take out the CPU, rotate it and try to plug it back in a different position, is that working?

No, it's not working since CPU should be placed in a specific direction to match the pins on the motherboard.

Explain what overclocking is?

Overclocking is the action of increasing a component's clock rate, running it at a higher speed than it was designed to run.

What are pins on a PCI/PCI-e card and what are they used for?

The pins on a PCI/PCI-e card are designed for plugging the card into the slot on the motherboard for connection.

Before PCI-e became a common standard, many graphics cards were using Accelerated Graphics Port (AGP), explain why.

It increases the speed at which machines can render graphics while using the system's resources more efficiently to assist in the acceleration of 3D computer graphics.

II. Command line interface

mkdir, touch, mv, cp, and ls

```
1 # Create a file named test
2 touch test
3 # Move test to dir/test.txt, where dir is a new directory
4 mkdir dir
5 mv test dir/test.txt
6 # Copydir/test.txttodir/test_copy.txt
7 cp dir/test.txt dir/test_copy.txt
8 # List all the files contained in dir
9 ls -a dir
```

grep

```
1 # List all the files from /etc containing the pattern 127.0.0.1
2 grep -rl "127.0.0.1" /etc
3 # Only print the lines containing your username and root in the
   file /etc/passwd
4 grep -rE 'michaelhuang.*root|root.*michaelhuang' /etc/passwd
```

find

```
1 # List all the files from /etc that have been accessed less than
   24 hours ago
2 find /etc/ -atime -1
3 # List all the files from /etc whose name contains the pattern "
   netw"
4 find /etc/ -name '*netw*'
```

redirection

- > redirect standard output into a file named on the right side of > and override its original content
- >> redirect standard output into a file named on the right side of >> and append after its original content
- <<< redirect the content on the right side of <<< as standard input of the command on the left
- >&1 redirect standard output into standard output
- 2>&1 > redirect standard error into standard output and then redirect standard output into a file named on the right side of the second >
- `tee` redirect standard input to a file named on the right side of `tee` and print it to standard output

xargs and |

- `xarg` build and execute command lines from standard input
- | pipe standard output of the command on the left side of | into the command on the right side of | as standard input.

head and tail

- `head` output the first part of files
- `tail` output the last part of files
- `tail -f` “live display” a file as new lines are appended

mirror

- `ps` display information about a selection of the active processes
- `top` display Linux processes and provides a dynamic real-time view of a running system
- `free` display the total amount of free and used physical and swap memory in the system, as well as the buffers and caches used by the kernel
- `vmstat` report information about processes, memory, paging, block IO, traps, disks and cpu activity

shell

- `sh` the original Bourne shell
- `bash` the GNU Bourne-again shell, which is mostly Bourne-compatible, mostly POSIX-compatible, and has other useful extensions
- `csh` the shell written by C programming language, which is not Bourne-compatible
- `zsh` the shell boasts extra and unusual features, which is almost Bourne-compatible

\$

- `$0` expand to the name of the shell or shell script
- `$1` the first argument to the script
- `$?` exit code from the previous command
- `$!` PID of last background command

PS3

`PS3` environment variable defining the prompt message in a shell script to display a user-friendly prompt instead of #?

```
1 #!/bin/sh
2 PS3="Select a day (1~3): "
3 select i in Mon Tue Wed
4 do
5     echo -ne "You choose: "
6     case $i in
7         Mon) echo "Monday";;
8         Tue) echo "Tuesday";;
9         Wed) echo "Wednesday";;
10        *) echo nothing;;
11    esac
12    break
13 done
```

iconv

`iconv` reads in text in one encoding and outputs the text in another encoding, which is useful especially in converting files with different encoding on different platforms.

\$temp

- `#${temp}` the number of characters in variable `$temp`
- `${temp%%word}` characters excluding the matching pattern `word` in variable `$temp`
- `${temp/pattern/string}` characters whose matching pattern `pattern` are replaced by `string` in variable `$temp`

files and directories

Files in Unix System are organized into multi-level hierarchy structure known as a directory tree. At the very top of the file system is a directory called “root” which is represented by a “/”. All other files are “descendants” of root.

- / the root of the filesystem tree
- /lib contains system libraries needed by the binary file in /bin, and some critical files such as kernel modules or device drivers
- /usr/lib stores the required libraries and data files for programs stored within /usr or elsewhere.
- /srv contains data for services provided by the system
- sbin usually contains files for system maintenance or administrative tasks
- /bin stands for “binaries” and contains certain fundamental utilities, such as ls or cp, which are generally needed by all users
- /mnt stands for “mount”. Contains filesystem mount points. These are used, for example, if the system uses multiple hard disks or hard disk partitions. It is also often used for remote (network) filesystems, CD-ROM/DVD drives, and so on
- /usr/src contains system-related source files
- /media default mount point for removable devices, such as USB sticks, media players, etc
- /dev stands for “devices”, which contains file representations of peripheral devices and pseudo-devices
- /boot contains all the files that are required for successful booting process
- /usr/bin this directory stores all binary programs distributed with the operating system not residing in /bin, /sbin or (rarely) /etc
- /proc procfs virtual filesystem showing information about processes as files.
- /opt contains subdirectories for optional software packages
- /vmlinuz linux kernel executable
- /etc contains system-wide configuration files and system databases
- /usr/share contains architecture-independent shareable text files
- /sys an interface to the kernel that provides kernel-view information and configuration settings
- /var a short for “variable”, which is a place for files that may change often – especially in size, for example e-mail sent to users on the system, or process-ID lock files.
- initrd.img a link to the latest installed initrd (i.e., Linux initial RAM disk)

shell script game

```
1 #!/bin/sh
2 ans=$((RANDOM)%100)
3 guess=100
4 while true ; do
5     # check whether input is a number
6     while true ; do
7         read -p "Guess a number: " guess
8         if [[ $guess =~ ^[0-9]+$ ]] ; then
9             break
10        fi
11    done
12    # compare the guess with the answer
13    if [ $ans -lt $guess ] ; then
14        echo "Smaller"
15    elif [ $ans -gt $guess ] ; then
16        echo "Larger"
17    else
18        echo "You win! The number is" $ans
19        break
20    fi
21 done
```