

# Chapter 1 Introduction: Lab 1

A/Prof Jian Zhang

Tutor: Xiaoshui Huang

Exercise 1: Understand the Linux basic command line.

Command	Description
cat [filename]	Display file's contents to the standard output device (usually your monitor).
cd /directorypath	Change to directory.
chmod [options] mode filename	Change a file's permissions.
chown [options] filename	Change who owns a file.
clear	Clear a command line screen/window for a fresh start.
cp [options] source destination	Copy files and directories.
date [options]	Display or set the system date and time.
df [options]	Display used and available disk space.
du [options]	Show how much space each file takes up.
file [options] filename	Determine what type of data is within a file.
find [pathname] [expression]	Search for files matching a provided pattern.
grep [options] pattern [filename]	Search files or output for a particular pattern.
kill [options] pid	Stop a process. If the process refuses to stop, use kill -9 pid.
less [options] [filename]	View the contents of a file one page at a time.
ln [options] source [destination]	Create a shortcut.
locate filename	Search a copy of your filesystem for the specified filename.
lpr [options]	Send a print job.

ls [options]	List directory contents.
man [command]	Display the help information for the specified command.
mkdir [options] directory	Create a new directory.
mv [options] source destination	Rename or move file(s) or directories.
passwd [name [password]]	Change the password or allow (for the system administrator) to change any password.
ps [options]	Display a snapshot of the currently running processes.
pwd	Display the pathname for the current directory.
rm [options] directory	Remove (delete) file(s) and/or directories.
rmdir [options] directory	Delete empty directories.
ssh [options] user@machine	Remotely log in to another Linux machine, over the network. Leave an ssh session by typing <b>exit</b> .
su [options] [user [arguments]]	Switch to another user account.
tail [options] [filename]	Display the last <i>n</i> lines of a file (the default is 10).
tar [options] filename	Store and extract files from a tarfile (.tar) or tarball (.tar.gz or .tgz).
top	Displays the resources being used on your system. Press q to exit.
touch filename	Create an empty file with the specified name.
who [options]	Display who is logged on.

Exercise 2: Write a program to send some words from memory to monitor and then display it.

```
#include <stdio.h>
void help()
{
    printf("The program should have one input parameter,\n");
    printf("the parameter is the content of display.\n");
    printf("Usage: exe content\n");
    printf("Example: exe \" I am the first program!\"\n");
}
int main(int argc, char **argv){
    if (argc != 2) help();
    else
    {
        printf("The display content is:\n");
        printf("%s\n",argv[1]);
    }
}
```

```

    return 0;
}

```

Exercise 3: To help you understand the I/O operation and file-system manipulation:

Read a text file from a hard disc to a memory. This is done by 1) execute a program through its command line, 2) the program includes open a text file and read contents of the text file 3) input the contents to a specific location the will display them.

```

#include <stdio.h>

void help()
{
    printf("The program should have one input parameter,\n");
    printf("the parameter is the document to be read.\n");
    printf("Usage: exe text_file_path\n");
    printf("Example: exe txt_file_path\n");
}

int main(int argc, char**argv)
{
    if (argc != 2) help();
    else {
        // open a text file
        FILE* file=fopen(argv[1], "r");
        char strLine[256];
        char ch;
        if (file != NULL) {
            fgets(strLine,256,file); //read a line from the text file.
            printf("%s\n",strLine);

            //read two characters
            fread(&ch, sizeof(char), 1, file);
            printf("%c",ch);
            fread(&ch, sizeof(char), 1, file);
            printf("%c", ch);

            //read the remaining content of the line.
            fgets(strLine, 256, file); //read a line from the text file.
            printf("%s\n", strLine);
        }
    }
}

```