ICSI333 System Fundamentals

Note: Students are expected to start the activities as soon as the description is available and seek feedback as needed. Although some activities are not graded for credit, they are contiguous study of the lecture or used as stepping-stones for the projects. Skipping any activities would impact the learning significantly.

Objectives:

- Practice files and basic file functions

Reading:

- Lecture notes

Submission (5 points):

- All required C programs must be submitted on Duifene on time.

Instructions:

Task #1. Study the following examples.

Example 1: Create a file and write data to it.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
	FILE *fptr;
	char name[20];
	int age;
	float salary;

	fptr = fopen("record.txt", "w");
	if (fptr == NULL){
		exit(-1);
	}
	printf("Enter the name \n");
	scanf("%s", name);
	fprintf(fptr, "Name = %s\n", name);

	printf("Enter the age\n");
	scanf("%d", &age);
	fprintf(fptr, "Age = %d\n", age);

	printf("Enter the salary\n");
	scanf("%f", &salary);
	fprintf(fptr, "Salary = %.2f\n", salary);

	fclose(fptr);
}
```

Example 2: Append to the end of a file with command-line arguments.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]){
	char name[50];
	int age;
	float salary;
```

```
	FILE *fptr;

	if(argc==4){
		strcpy(name, argv[1]); //array is not assignable
		age = atoi(argv[2]);
		salary = atof(argv[3]);
	} else {
		perror("\n");
		exit(-1);
	}


	fptr = fopen("record.txt", "a");
	if (fptr == NULL){
		exit(-1);
	}

	fprintf(fptr, "Name = %s\n", name);
	fprintf(fptr, "Age = %d\n", age);
	fprintf(fptr, "Salary = %.2f\n", salary);

	fclose(fptr);
}
```

Example 3: Read from a file.

```
#include <stdio.h>

int main(){
	FILE *fp = fopen("record.txt", "r");
	int ch = getc(fp);          //return value of getc is integer
	while (ch != EOF){          //or use feof(fp) function
		putchar(ch);            //display the char
		ch = getc(fp);
	}

	fclose(fp); return 0;
}
```

Task #2. Write a program that extracts words from a text file.

Write a program that reads a line from a file starting from the first line, prints the line by displaying each word in its own line and repeats until the last line is processed. Any whitespace characters can be used as the delimiter of the words. You can test the program using the same program as a text file.

Task #3. Study the following examples and answer the question [optional]

Function designs can be found in the Linux Programmer's Manual using `man` command. For example, `man fopen` will display the manual for C function `fopen`. You may need to enter `man 3 fopen` explicitly to go to section 3 of the manual for the C standard functions. Can you study more functions from the manual?

.

Linux Basic Commands

`ls [option(s)] [file(s)]`	If you run ls without any additional parameters, the program will list the contents of the current directory in short form. `-l` detailed list `-a` displays hidden files
`cp [option(s)] sourcefile targetfile`	Copies sourcefile to targetfile. `-I` waits for confirmation, if necessary, before an existing targetfile is overwritten `-r` copies recursively (includes subdirectories)
`mv [option(s)] sourcefile targetfile`	Copies sourcefile to targetfile then deletes the original sourcefile. `-b` creates a backup copy of the sourcefile before moving. `-I` waits for confirmation, if necessary, before an existing targetfile is overwritten.
`rm [option(s)] file(s)`	Removes the specified files from the file system. Directories are not removed by rm unless the option `-r` is used. `-r` deletes any existing subdirectories `-I` waits for confirmation before deleting each file
`cd [options(s)] [directory]`	Changes the current directory. cd without any parameters changes to the user's home directory.
`mkdir [option(s)] directoryname`	Creates a new directory.
`rmdir [option(s)] directoryname`	Deletes the specified directory, provided it is already empty.
`cat [option(s)] file(s)`	The cat command displays the contents of a file, printing the entire contents to the screen without interruption. `-n` numbers the output on the left margin
`cal`	Displays the calendar of the current month.
`date`	Displays current time and date.
`whoami`	Reveals the user who is currently logged in.
`whatis`	Gives a one-line description about the command. It can be used as a quick reference for any command.

`man`	Manual Pages, for more detailed information, Linux provides man pages and info pages. To see a command's manual page, man command is used.
`pwd`	Prints the absolute path to current working directory.
`vi`	A text editor for Linux operating system.
`gedit`	The default GUI text editor in the Ubuntu operating system.
`emacs`	Another text editor for Linux operating system.

Vi commands:

To save and quit:

Commands	Action
:wq	Save and quit
:w	Save
:q	Quit
:w fname	Save as fname
ZZ	Save and quit
:q!	Quit discarding changes made
:w!	Save (and write to non-writable file)

More commands:

Copy-pasting within the terminal: Ctrl+Shift+C/V

Resources and Credits:

- Teaching materials from Professor Kuperman at UAlbany
- Vi reference: https://www.ele.uri.edu/faculty/vetter/Other-stuff/vi/vi-quick-ref.pdf