



SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Established in collaboration with MIT

Computer System Engineering
50.005

Week 1: Lab 4 (25 marks)

Objective: File Operation in Shell Interface

Contact us

chuadhry@mymail.sutd.edu.sg

eyasu_chekole@mymail.sutd.edu.sg

shuailong_liang@mymail.sutd.edu.sg

hao_zhang@mymail.sutd.edu.sg

The Objective of this lab

- In this lab, we will continue our work on lab1 by implementing file operation in Shell Interface using Java or C language.
- We extend the Shell Interface with several file operation methods.

Tasks to Perform!

- In this lab your code should handle four main tasks:
 1. Implement functions to create, delete display a file
 2. Implement function to list a directory
 3. Implement function to find files under current directory and subdirectories
 4. Implement function to list subdirectories and files in a tree structure

Where to start!

- Open your eDimension and download the report for lab4
- Decide which language do you prefer based on your background
 - Java or C language
- Read the tasks one by one and use the help code provided in the report and the starting code in eDimention
- Don't hesitate to ask for help from the teachers in the lab!

Create

File and folders under current directory:

```
a.out  
abc  
FileOperation$1.class  
FileOperation$2.class  
FileOperation$3.class  
FileOperation.class  
FileOperation.java  
lab1_good_sol  
Q4.c
```

When we type in the following command:

jsh>create 1.txt or csh>create 1.txt

```
.DS_Store  
1.txt  
a.out  
abc  
FileOperation$1.class  
FileOperation$2.class  
FileOperation$3.class  
FileOperation.class  
FileOperation.java  
lab1_good_sol  
Q4.c
```

Create

Java:

```
File file = new File(File dir, String name);  
file.createNewFile();
```

C:

```
FILE *fp;  
fp = fopen(fileName, "w+");
```

Delete

File and folders under current directory:

```
.DS_Store  
1.txt  
a.out  
abc  
FileOperation$1.class  
FileOperation$2.class  
FileOperation$3.class  
FileOperation.class  
FileOperation.java  
lab1_good_sol  
Q4.c
```

When we type in the following command:

jsh>delete 1.txt or csh>delete 1.txt

```
a.out  
abc  
FileOperation$1.class  
FileOperation$2.class  
FileOperation$3.class  
FileOperation.class  
FileOperation.java  
lab1_good_sol  
Q4.c
```

Delete

Java:

```
File file = new File(File dir, String name);  
file.delete();
```

C:

You can use one of the two following functions to do that:

1- **unlink** function:

```
int unlink(const char *filename);
```

2- **remove** function:

```
int remove(const char *filename);
```


Display

When we type in the following command:

```
jsh>display test.txt csh>display test.txt
```

The content inside “test.txt” will be displayed:

Hello.

This is the content inside test.txt file.

Display

Java:

```
File file = new File(File dir, String name);  
FileReader fileReader = new FileReader(file);  
BufferedReader in = new BufferedReader(fileReader);  
String line;  
  
while((line = in.readLine()) != null){  
    System.out.println(line);  
}  
  
in.close();
```

Display

C:

You can use different ways to do that, the easiest one is to read the file content line by line or char by char, either by `scanf` function or `fgetc` function:

```
FILE *fp;  
fp = fopen(filename, "r");  
char ch;  
while((ch = fgetc(fp)) != EOF)  
    printf("%c", ch);
```

List

File and folders under current directory:

When we type in the following command:

`jsh>list` or `csh>list`

The files under current directory will be printed out:

```
a.out
abc
FileOperation$1.class
FileOperation$2.class
FileOperation$3.class
FileOperation.class
FileOperation.java
lab1_good_sol
Q4.c
```

List

Show property of files

When we type in the following command:

`jsh>list property` or `csh>list property`

The files under current directory will be printed out:

Week1	Size: 4096	Last Modified: Mon Jan 26 13:10:47 SGT 2015
Week2	Size: 4096	Last Modified: Sun Jan 18 21:09:22 SGT 2015
Week3	Size: 4096	Last Modified: Thu Feb 05 16:43:57 SGT 2015
Week5	Size: 0	Last Modified: Thu Feb 12 16:16:27 SGT 2015

List

Sort the file list by different property

When we type in the following command:

```
jsh>list property time
```

The files under current directory will be printed out:

Week2	Size: 4096	Last Modified: Sun Jan 18 21:09:22 SGT 2015
Week1	Size: 4096	Last Modified: Mon Jan 26 13:10:47 SGT 2015
Week3	Size: 4096	Last Modified: Thu Feb 05 16:43:57 SGT 2015
Week5	Size: 0	Last Modified: Thu Feb 12 16:16:27 SGT 2015

List

Java:

Get file list:

File dir;

```
File[] list = dir.listFiles();
```

```
for (File file: fileList)
```

```
{ System.out.println(file.getName());}
```

Get file property:

File file;

```
file.getName();
```

```
file.length();
```

```
new Date(file.lastModified());
```

Sort file list:

Function is provided:

```
private static File[] sortFileList(File[] list, String sort_method);
```

List C

- `#include <dirent.h>` // header to perform operations on dir
- `opendir()` to open a directory
- `readdir()` to read a directory
- `closedir()` to close a directory

List C

- create DIR * which is simply a pointer to a directory.
- DIR *pdir
- struct dirent *pent
- struct dirent * which is basically used when reading a directory
- function readdir() will return a dirent
- pdir = opendir (".");
- while (pent = readdir (pdir)) // while there is still something in the directory to list
- printf ("%s\n", pent->d_name);
- closedir (pdir);

List C, continue ...

Get file property:

stat function:

```
#include<sys/stat.h>
```

```
int stat(const char *restrict path, struct stat *restrict buf);
```

Details:<http://pubs.opengroup.org/onlinepubs/009695399/functions/stat.html>

then use the struct fields to find each property like:

st_size, st_mtime and etc...

List C, continue...

Sort file list:

```
#include<dirent.h>
```

```
int scandir(char *dirp, struct dirent ***namelist, int (*filter)(struct dirent *), int (*compare)(struct dirent **, struct dirent **));
```

```
and int alphasort(const struct dirent **d1, const struct dirent **d2);
```

<http://pubs.opengroup.org/onlinepubs/9699919799/functions/alphasort.html>

<http://man7.org/linux/man-pages/man3/scandir.3.html>

Find

When we type in the following command:

```
jsh>find .java csh>find .java
```

All files with “.java” substring under current directory and subdirectories will be shown:

```
C:\CSE_Lab\src\Week1\SimpleShell.java
```

```
C:\CSE_Lab\src\Week2\MergeSortThreaded.java
```

```
C:\CSE_Lab\src\Week2\MultiThread.java
```

```
C:\CSE_Lab\src\Week3\Bank.java
```

```
C:\CSE_Lab\src\Week3\BankImpl.java
```

```
C:\CSE_Lab\src\Week3\TestBank.java
```

```
C:\CSE_Lab\src\Week5\FileOperation.java
```

Find: Java

In order to find files in current directory and its subdirectories, we need to implement a recursive function.

Function to get path of a file:

```
File file;  
file.getAbsolutePath();
```

Function to check whether a file is a directory(folder):

```
File file;  
file.isDirectory();
```

Find: C

In order to find files in current directory and its subdirectories,
You need to compare your files names with your substring,
using **strstr** function:

```
#include<string.c>  
char *strstr(const char *haystack, const char *needle);
```

Tree

When we type in the following command:

`jsh>tree` or `csch>tree`

All files under current directory and its subdirectories will be shown in a tree structure:

Week1

|-SimpleShell.java

Week2

|-data

|-input_1.txt

|-input_2.txt

|-MergeSortThreaded.java

|-MultiThread.java

Week3

|-Bank.java

|-BankImpl.java

|-TestBank.java

Week5

|-FileOperation.java

Tree

Like the **find** function, the **tree** function should also be recursive. (show current directory and subdirectories)

We should be able to control the maximum level of subdirectories to be shown.

When we type in the following command:

```
jsh>tree 1 or csh>tree 1
```

The top level files will be shown:

Week1

Week2

Week3

Week5

Tree

We should be able to control the maximum level of subdirectories to be shown.

When we type in the following command:

```
jsh>tree 2 or csh>tree 2
```

The files in top 2 levels will be shown:

Week1

- |-SimpleShell.java

Week2

- |-data

- |-MergeSortThreaded.java

- |-MultiThread.java

Week3

- |-Bank.java

- |-BankImpl.java

- |-TestBank.java

Week5

- |-FileOperation.java

Tree

We should be able to control the **maximum level** of subdirectories to be shown **based on specific property** like Size, Time, Name ...

When we type in the following command:

jsh>tree **2 time** or csh>tree **2 time**

The files in top 2 levels will be shown based on last time modified:

```
— 1.txt
— Week1
  — SimpleShell.c
— Week2
  — data
— Week3
  — Bank.c
  — BankTmp1.c
  — TestBank.c
— Week4
  — FileOperation.c
Q4
```

Question 1, 2, 3 & 4

- Complete the program and upload the Java file or C file(for Q1 to Q4) to eDimension Friday **10 March 2016 11:59 PM**