DT271C - Seminar 4 - Filesystems and I/O

Introduction

The purpose of this Seminar is extend the ideas from seminar 1 to show how the filesystem works for an operating system and also showcase input/output-operations.

In these tasks you will extend the "Java Command shell" that you were given in seminar 1. You can either based the implementation on your previous work or start over again from the base code (CommandShell.java).

Read the complete document before you start with the specific tasks.

Good luck!

1 Programming tasks

In this seminar we have three major tasks.

1.1 Task 1 - filedump

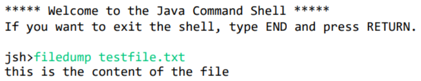
In this task you are going to add a new command that you can use from the Java Command shell. The name of the new command is “filedump” and it will take a filename as parameter. The command shall show the content of a given filename.

filedump filename

For example: “filedump testfile.txt”

Filedump command shall then printout the content of the testfile.txt in the terminal.

Example usage:



```
***** Welcome to the Java Command Shell *****
If you want to exit the shell, type END and press RETURN.

jsh>filedump testfile.txt
this is the content of the file
```

Figure 1: filedump command used.

In the above example the testfile.txt is already created in notepad with the content *“this is the content of the file”*.

1. The filedump command shall work with any filename and not only the testfile.txt
2. The filedump command shall print the content of the file in the terminal.
3. You are not allowed to create a wrapper around any Windows or Linux command like, “cat” or similar.
4. You don’t have to support a complete path for the file.
5. Implement basic error handling, for example if the file does not exist.

1.2 Task 2 - copyfile

In this task you are going to add a new command that you can use from the Java Command shell.

The name of the new command is “copyfile” and it will take a source filename and destination filename. The command will copy the content of the source file to the destination file.

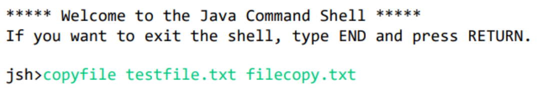
copyfile sourcefile destinationfile

For example: “copyfile testfile.txt filecopy.txt”

Copyfile shall now copy the content of testfile.txt to filecopy.txt.

Example usage:

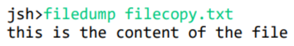
And then we can check the content of the destination file:



```
***** Welcome to the Java Command Shell *****
If you want to exit the shell, type END and press RETURN.

jsh>copyfile testfile.txt filecopy.txt
```

Figure 2: copyfile command used.



```
jsh>filedump filecopy.txt
this is the content of the file
```

Figure 3: filedump of copied file.

1. The copyfile command shall work with any filenames and not only testfile.txt and filecopy.txt
2. The copyfile command shall at least work with file content in form of “text” data
3. You are not allowed to create a wrapper around any Windows or Linux command like, “cp” or similar.
4. You don’t have to support a complete path for the file.
5. Implement basic error handling, for example if the file does not exist.

1.3 Task 3 - Another command

Add support for a new command in the command shell that has something to do with file systems or file manipulation. You can decide on you own what the command will do. Some suggestions are: “createfile”, “renamefile”, “appendfile”, “createdir”.

Also describe the new command, in either a text-file or as Java-doc and upload this with the source code on canvas. Write at least 10 lines describing the new command and how it works.

1. Add support for a new command
2. The copyfile command shall at least work with file content in form of “text” data
3. The new command shall functionality related to file systems or file manipulation
4. The new command shall be called with at least one argument.
5. You are not allowed to create a wrapper around any Windows or Linux command.