Frequently Asked Questions: Programming Assignment 2

How should the program be executed?

In the following examples, FILENAME stands for the name of the input file which contains the list of commands.

If you used C or C++, we should be able to execute your program by entering: ./access FILENAME, where access is the name of the executable. To make the name of the executable "access", specify the -o access option when compiling using gcc or g++.

If you used Java, we should be able to execute your program by entering: java access FILENAME. Your main class should be named "access" and the compiled file name for it should be "access.class".

If you used Python, we should be able to execute your program by entering: python access.py FILENAME, or python3 FILENAME, where access.py is the name of the file that contains your main function.

How do you parse the command string?

Any way is fine. (As long as it works.) A suggested method is to "tokenize" the command line into separate word strings and go from there. For example, if the command is “write script echo Hello World!”, you could do the following in Java:

```
String command = "write script echo Hello World!"; // Read from file instead of hardcoded.
int maxNumTokens = 3;
...
String[] tokens = command.split(" ", maxNumTokens);
...
// The tokens variable is a string array of length 3 that contains the tokens from the
command string.
// Now, tokens[0] contains the string "write"
// and tokens[1] contains the string "script"
// and tokens[2] contains the string "echo Hello World!"
```

For more information regarding the String.split(String,int) method, see this link.
Note: if you are using Python, look into the cmd module.

Should the program handle exceptions related to incorrect syntax?

No. To simplify the assignment it is assumed that there are no syntax errors. Thus, your programs will be tested only with correct instructions (no typos, no extra arguments, no extra long filenames, etc.).

Do you need to check for illegal characters in usernames, passwords, filenames, and/or group names?

No. To simplify the assignment it is assumed that usernames, passwords, filenames, and group names will contain only valid characters as specified in the assignment handout.

A username, group name, or filename may consist of up to 30 ASCII characters except for a forward slash ('/'), colon (':'), or any of the following white space characters: carriage return, form feed, horizontal tab, new line, vertical tab, and space.

A password may consist of up to 30 ASCII characters except for the following white space characters: carriage return, form feed, horizontal tab, new line, vertical tab, and space.

How should the reserved filenames be handled?

The following files are reserved: friends.txt, audit.txt, lists.txt, and pictures.txt. Therefore, any command that references one of these files should result in an error message that is BOTH displayed to the terminal and logged to the audit.txt. For example, if the command “postpicture pictures.txt” was encountered, your program should display and log an error message such as "Error: invalid filename".

What format(s) should be used for the files?

All required and user-created files that are generated by your program should be in an ASCII text format.

The REQUIRED formats for pictures.txt, lists.txt, and audit.txt are given in the test cases. Note that the format for friends.txt is an EXAMPLE. You may organize the friends.txt file differently, but remember to describe the format you have used in the documentation.

User-created files should contain one line of text for each successful call to a write command that corresponds to the file. Again, examples are given in the test cases.

The formats for the files are included at the end of the line-by-line test cases.

When should the files be created? Overwritten? Appended?

audit.txt

This is the audit log file. It should be an ASCII text file. It should be created when your program first starts before any instruction is executed or when the very first entry to the log is written. It should be created in such a way that if an audit.txt file is already present in the filesystem (for example, from a previous run), the old file is overwritten. Each line thereafter should be appended to the file. Examples of audit.txt files are given in the test cases.

friends.txt

This is the ASCII file in which your program should store friends. It should be created when your program first starts before any instruction is executed or when the very first friend account is created. It should be created in such a way that if a friends.txt is already present in the filesystem (for example, from a previous run), the old file is overwritten. Each line thereafter should be appended to the file. An EXAMPLE format is given in the test cases.

lists.txt, pictures.txt

These should both be ASCII text files. Both should be created and written after the end command is read, but before your program terminates. They should be created in such a way that if a lists.txt and/or pictures.txt are already present in the filesystem (for example, from a previous run), the old files are overwritten. Each line thereafter should be appended to the files. Examples of the REQUIRED formats are given in the test cases.

All other user created files

Any other file should be created when the postpicture command is successfully executed for that file. Each file should be created in such a way that if a file by the same name is are already present in the file system (for example, from a previous run), the old files is overwritten. Each line thereafter should be appended.

Are usernames, passwords, filenames, and/or group names case sensitive?

YES! ALL friends, lists, and pictures are CASE SENSITIVE!

BOb and bob are two different friends. PICture and Picture are two different pictures.

File.txt and file.txt are two different filenames. Be careful if you are developing your program in an operating system (like any version of Windows) that uses case-insensitive filenames. Make sure that you test your program on the LCSEE Ubuntu Linux shell servers. Your program will be likely be tested to see if it differentiates between friends, lists, and/or pictures that differ only by case.