Test Plan

Matthew Fortier, Anthony Taylor
December 7, 2016

1 Introduction

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

2 The Shell

The shell is going to be final deliverable for this project, and will allow a user to execute all of the commands to follow. It is imperative that the shell works correctly before we test our written commands with it.

2.1 What to look for

The shell is going to be forking off processes when a command is entered in the shell. It is very important that the shell correctly and reliably forks processes, and waits for them to join before allowing the use to execute the next. In this case, it is a good idea to print the process id and parent id using `getpid()` and `getppid()` to the standard output, and then comparing the process id with the systems list of processes using `ps aux | less`. This way we are sure the processes are being created and destroyed properly.

It is also a good idea to make sure the error reporting of the shell is fully operational. Again, using pre-written bash commands, test to make sure the shell is returning all the correct errors for that command. For safety, test the shell with a handful of commands to be sure the shell returns the correct errors and success states.

3 Commands

3.1 cat x

The cat commands is an abbreviation of con**cat**enate and when successfully executed, prints the contents of x to the screen.

3.1.1 Test Cases

Command	Test Case	Expected	Action	Comments
cat test.txt	x is a file	Pass	Print contents of test.txt	Relative directory
cat /home/test.txt	x absolute directory	Pass	Print contents of /home/test.txt	Normal operation
cat hidden.txt	x does not exist	Fail	Print missing error	x must exist
cat	no arguments	Fail	Print argument error message	There needs to be one and only one argument
cat test1.txt test2.txt	More than 1 argument	Fail	Print argument error	There needs to be one and only one argument
cat /	x is a directory	Fail	Print directory error message	x cannot be a directory

3.1.2 Success Check

Check to see if the file printed to the screen correctly after each test.

3.2 cd *x*

Acronym for change directory. Changes to the specified directory.

3.2.1 Test Cases

Command	Test Case	Expected	Action	Comments
cd /	x is a absolute directory	Pass	Changes to specified directory	Normal operation
cd home	x relative directory	Pass	Changes to specified directory	Normal operation
cd	x is null	Pass	Change to home directory	Normal operation
cd /not-exist	Absolute directory does not exist	Fail	Print directory error message	Directory must exist
cd not-exist	Relative directory does not exist	Fail	Print directory error message	Directory must exist

3.2.2 Success Check

Check to make sure the program successfully changed the current directory.

3.3 df

Acronym for disk free. Print number of free logical blocks.

3.3.1 Test Cases

Command	Test Case	Expected	Action	Comments
df		Pass	Prints logical blocks	Normal operation
df x	Contains arguments	Pass	Prints logical blocks	Ignore arguments

3.3.2 Success Check

Check to make sure the correct information is sent to the screen, it is formatted correctly and the info looks accurate.

3.4 ls *x*

Acronym for list. When x is the name of a file, list the name of the file, with its extension, the file type, FLC and file size. When x is the name of a directory, then list the names of the entries (with extensions) within the directory along with FLC, file size and file type (file or directory). Also, list the âĂIJ.âĂİ (current directory) and âĂIJ..âĂİ (parent directory) entries.

3.4.1 Test Cases

Command	Test Case	Expected	Action	Comments
ls	List all files and extensions and all sub-directories	Pass	Prints all specified files and directories info	Normal operation
ls test.txt	List file details	Pass	Prints all specified file info	Normal operation
ls .	List current directory	Pass	List current directory	Normal operation
ls ..	List parent directory	Pass	List parent directory	Normal operation
ls /	List boot directory	Pass	List boot directory	Normal operation
ls /home/user/test.txt	List absolute filename	Pass	List absolute file info	Normal operation
ls test.txt	List relative filename	Pass	List relative file info	Normal operation
ls Desktop Home	Two directories	Fail	Print argument error	Normal operation
ls Desktop test.txt	Two arguments	Fail	Print argument error	Normal operation
ls test1.txt test2.txt	Two files	Fail	Print argument error	Normal operation

3.4.2 Success Check

Check output to make sure all files and directories are outputted, check to make sure it is formatted correctly, and make sure all information on each file and directory is correct and displayed.

3.5 mkdir *x*

Acronym for make directory. Create directory in current or under absolute directory.

3.5.1 Test Cases

Command	Test Case	Expected	Action	Comments
mkdir myDirectory	Create myDirectory under current directory	Pass	Creates myDirectory	Normal operation
mkdir /home/folder	Create folder under absolute directory	Pass	Creates folder	Normal operation
mkdir /home/exists	x exists	Pass	Print x exists message	Normal operation
mkdir folder1 folder2	2 arguments	Fail	Print too many arguments error	Only one argument is allowed

3.5.2 Success Check

Check to make sure the directory is created.

3.6 pbs

Acronym for print boot sector.

3.6.1 Test Cases

Command	Test Case	Expected	Action	Comments
pbs	Normal	Pass	Prints logical blocks	Normal operation
pbs x	Contains arguments	Pass	Prints boot sector	Ignore arguments

3.6.2 Success Check

Check to make sure all the information is printed to the screen, check if the accuracy of the info, and make sure it is formatted correctly.

3.7 pfe *x y*

Acronym for print fat entries. Print the 12-bit FAT entry values representing logical sectors x to y.

3.7.1 Test Cases

Command	Test Case	Expected	Action	Comments
pfe 2 8	x > y and x > 2	Pass	Prints FAT entry	Normal operation
pfe	No arguments	Fail	Print no arguments error	Normal operation
pfe 2	Missing argument	Fail	Print missing argument error	Normal operation
pfe 2 8 8	Too many arguments	Fail	Prints too many arguments error	Normal operation
pfe 1 8	x < 2	Fail	Print x error	x must be >= 2
pfe 4 3	x > y	Fail	Print error	x must be < y

3.7.2 Success Check

Check to make sure all the information is printed to the screen, check if the accuracy of the info, and make sure it is formatted correctly.

3.8 pwd

Print working directory.

3.8.1 Test Cases

Command	Test Case	Expected	Action	Comments
pwd	Normal	Pass	Prints working directory	Normal operation
pwd x	Contains arguments	Pass	Prints working directory	Ignore arguments

3.8.2 Success Check

Check to make sure that the working directory provided is actually the correct output.

3.9 rm *x*

Acronym for remove.

3.9.1 Test Cases

Command	Test Case	Expected	Action	Comments
rm test.txt	Relative pathname	Pass	Removes file	Normal operation
rm /home/test.txt	Absolute pathname	Pass	Removes file	Normal operation
rm	No arguments	Fail	Print argument error	x must be a directory
rm /home	x is a directory	Fail	Print directory error	x must be a directory
rm test.txt	test.txt does not exist	Fail	Prints file not exist error	File must exist
rm test1.txt test2.txt	Too many arguments	Fail	Print argument error	Only one argument is accepted

3.9.2 Success Check

Check to make sure the file was removed successfully, and that operation did not alter any others files or directories.

3.10 rmdir *x*

Acronym for remove directory.

3.10.1 Test Cases

Command	Test Case	Expected	Action	Comments
rmdir /home/user	Absolute pathname and empty	Pass	Removes directory	Normal operation
rmdir user	Relative pathname and empty	Pass	Removes directory	Normal operation
rmdir	No arguments	Fail	Prints argument error	Must contain pathname
rmdir /home/user	Directory is not empty	Fail	Prints empty error	Cannot remove filled directory
rmdir test.txt	x is a file	Fail	Prints file error	Must contain pathname

3.10.2 Success Check

Check to make sure the directory is removed correctly and that the removal process did not alter any other directories.

3.11 touch *x*

Creates file x.

3.11.1 Test Cases

Command	Test Case	Expected	Action	Comments
touch test.txt	Relative pathname	Pass	Creates test.txt in current directory	Normal operation
touch /home/test.txt	Absolute pathname	Pass	Creates test.txt in absolute directory	Normal operation
touch test.txt	test.txt exists	Fail	Print exists error	File must not exist
touch	No arguments	Fail	Print arguments error	Must have argument

3.11.2 Success Check

Check to make sure the file is created and that the operation did not interfere with any other files and directories.