

Assumptions

Command	Options	Note
rm [Options] [FILES]	FILES can be file or folder -d: delete empty directory -r: traverse recursively to delete the folder contents, if the folder is empty afterwards, delete the folder as well options can go together -rd / -dr	For file and directory error , only error message is written, no exception is thrown so that valid files can still be deleted File needs WRITABLE_PERMISSION to be delete
echo [ARG]	Arguments are separated by spaces, space is not considered an argument unless in a string argument (for example " ")	Always terminates by a newline on the standard output With no argument, echo writes a new line
paste [FILES]	FILES can be none, one or several file names. If FILE is not specified, data is read from stdin. FILE can be substituted with "-" for stdin	If FILES are not found or are directories, PasteException will be raised
sed REPLACEMENT FILE	FILE can be only one. If FILE is not specified, data is read from stdin. FILE CANNOT be substituted with "-" for stdin	If FILES are not found/are directories/does not have read permission, SedException will be raised
diff [Options] [FILES]	FILES can be either FILE1 FILE2 or DIR1 DIR2. FILE1 & FILE2 can be substituted with "-" for stdin, while DIR1 & DIR2 cannot be replaced	If FILES are not found or FILE1/FILE2 are directories or DIR1/DIR2 are files, DiffException will be raised If more than 2 FILES/ 2 DIRS are given, DiffException will be raised If 1 FILE and 1 DIR are given, DiffException will be raised
grep [-i] [-c] PATTERN [FILES]	FILES can be none, one or several file names. If FILE is not specified, data is read from stdin. FILE CANNOT be substituted with "-" for stdin	If FILES are not found, GrepException will be raised
wc [-clw] [FILES]	If FILES are not specified, use stdin	Only the fileName will be in the input not the path
cd PATH	PATH: relative or absolute directory path, platform	Raise exception for invalid path, file not exist

	independent Directory need to have READABLE_PERMISSION to be cd	
case1: cp SOURCE_FILE DESTINATION_FILE case2: cp [FILES] DIRECTORY		In case 1, content of source file will overwrite content of destination file In case 1, if either of them are not found, exception will be thrown In case 2, if a file in [FILES] is not found. An error message will be print out and execution will continue In case 2, if DIRECTORY is not found or is not a directory, exception will be thrown
cut [Option] [LIST] [FILES]	FILES can be none, one or several file names. If FILE is not specified, data is read from stdin. FILE can be substituted with “-” for stdin	If FILES are not found, CutException will be raised If LIST is out of range, CutException will be raised
ls [-d] [FOLDERS] [-R]		If FOLDERS are not found, LsException will be raised
sort [-nrf] [FILES]	-n If specified, treat the first word of a line as a number. -r - Sort in reverse order. -f - Convert all lowercase characters to their uppercase equivalent before comparison, that is, perform case-independent sorting. If FILES is not specified, sort from stdin instead	
find FOLDERS -name [filename]	flag -name always need to be there if no folders appear, current folder will be use to find	this command does not only find files but also folders
case1: mv [-n] SOURCE TARGET case2: mv [-n] [SOURCE] ... DIRECTORY		case2: if any source has some error (file not found, etc), error will be written as message so that other valid sources can be moved
exit		Throws exception if more than zero arguments are passed to the command.

Miscellaneous notes

“--” is considered to be an invalid option for most of the commands

The program stops when it reaches EOF, or when the user runs exit command.