**IS 340 – Operating Systems**

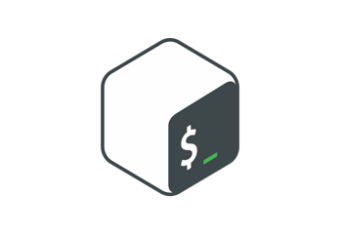
**HOP06 – BASH – Command-Line Parsing and Expansion**

7/14/2019 Developed by Kevin Wang

6/17/2020 Reviewed by Kim Nguyen

02/14/2022 Reviewed by Ken Ling

School of Technology & Computing (STC) @ City University of Seattle (CityU)

**Before You Start**

* This exercise assumes that the user is working with the Linux Distribution in MODS 1 and MOD 2
* All commands and code discussed in this exercise will run in the Ubuntu console.
* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below and experiment in the Ubuntu console and try to solve the problem yourself.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Parse command line with quoting, brace, tilde, variable, and arithmetic expansions
* Understand command substitution, word splitting, and pathname expansion
* Catch parsing options

**Resources**

# Linux command line: bash + utilities

<https://ss64.com/bash/>

* Nano/Basics Guide

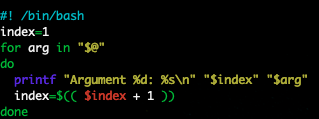
<https://wiki.gentoo.org/wiki/Nano/Basics_Guide>

**Preparation**

1. Create a new script via the command below:

>>> nano ShowArgument.sh

Then, type the following script to the file:



Note:

This script can print out different arguments followed by a command, which can help us to see how the bash script parse the arguments.

1. Hit the control + x key to quit and save the file.
2. Run a test by typing the following command:

>>> bash ShowArguments.sh first second ‘third argument’

You are supposed to see the following result



Note: please make sure you run the script successfully and get the correct output since it will be used for the next step.

**Expansions**

1. Testing the quoting style for command arguments by typing the following command:

>>> bash ShowArguments.sh argument1 argument2 ‘I am argument 3’

>>> bash ShowArguments.sh ‘This is how to include “double quotes”’

>>> bash ShowArguments.sh “This is how to include ‘single quotes’”



Note: please type the command by yourself since Word uses different characters for quotes.

1. Testing the brace expansion by typing the following commands:

>>> bash ShowArguments.sh {first,second,third}



>>> bash ShowArguments.sh {{1..5},{a..d}}



>>> bash ShowArguments.sh {1..3}{a..b}



>>> bash ShowArguments.sh {01..10..2}



>>> bash ShowArguments.sh {A..G..2}



1. Testing the tilde expansion by typing the following commands:

>>> bash ShowArguments.sh ~

>>> bash ShowArguments.sh ~root



Note: using the ~ + user’s name to get the users’ home directory. The username in this example is ubuntu.

1. Testing the variable expansion by typing the following commands:

>>> name=yourname

>>> bash ShowArguments.sh “${name}”



1. Testing the arithmetic expansion by typing the following commands:

>>> bash ShowArguments.sh “$(( 1 + 2 ))” “$(( (1 + 2) \* 2 ))”



**Understand command substitution, word splitting, and pathname expansion**

1. Command substitution is used to replace a command with its output. Testing it by typing the following commands:

>>> echo "Today is $( date +%Y-%m-%d )" > file.md

>>> cat file.md



1. The parameter will be split if we pass it as a variable without quotes. Test it by typing the following commands:

>>> names=”Kevin Arthur Evan”

>>> bash ShowArgurments.sh $names

>>> bash ShowArgurments.sh “$names”



Note: You can see the difference when we use double quotes for the variable.

1. If the command has a parameter that contains \*, ?, and [], bash will treat it as file patters. Testing it by typing the following commands:

>>> bash ShowArguments.sh Show\*

>>> bash ShowArguments.sh ?ile.md

>>> bash ShowArguments.sh [a-f]ile\*



1. The process substitution can be useful when we want to get the temporary file name for a command. Typing the following command to test:

>>> bash ShowArguments.sh <(ls -l) >(pr -Tn)



Note: this command allows the script getting a temporary input and output filename, which can allow programmers to read or write to.

**Parsing Options**

Bash script has a built-in command called getopts that can be used to parse options.

The basic syntax is: getopts OPTSTRING var

1. Create a OptionTest.sh file by typing the following command:

>>> nano OptionTest.sh

1. Type the following script in the file:



Note: we accept a, v, and c as legal flags. All other flags will be seen as invalid flags.

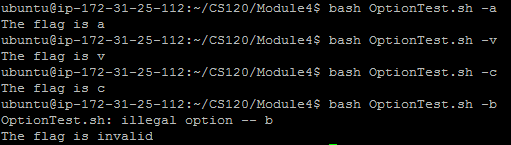
1. Hit the control + x key to quit and save
2. Test the script with the following commands:

>>> bash OptionTest.sh -a

>>> bash OptionTest.sh -v

>>> bash OptionTest.sh -c

>>> bash OptionTest.sh -b



**Instructions:**

**Change values like the range of integers in the first example of ShowArguments.sh. Print a screenshot and upload that into your document.**

**Change the name values in ShowArguments.sh. Print a screenshot and upload that into your document.**

**Change the name of the file in the ShowArguments.sh. Print a screenshot and upload that into your document.**

**Submit your Work to Brightspace**

Please upload all your files for this hands-on practice to the HOP assignment on Brightspace.