|  |  |
| --- | --- |
|  | Experiment no.7. |
| **Title** | **Command Line Arguments** |
|  |  |
| **Aim** | Write shell script to demonstrate Command Line Argument  i.Positional Parameters  ii.Using Flag  iii.Using Loops with $@ – Loop Constructs  iv. Use of $#,$\*,$$,$? And S! |
|  |  |
| **Hardware**  **Requirement** | Personal Computer |
|  |  |
| **Software**  **Requirement** | Linux Operating System(Ubuntu 20.04) , Shell-Interpreter  Nano or Vi or Vim or gedit text editor |
|  |  |
| **Theory** | **Command Line Arguments in Shell Script**  * Command line arguments are also known as positional parameters. * These arguments are specific with the shell script on terminal during the run time. * Each variable passed to a shell script at command line are stored in corresponding shell variables including the shell script name.   **Syntax:**  ./myshellscript.sh ARG1 ARG2 ARG3 ARG4 ARG5 ARG6 ARG7 ARG8 ARG9 ARG10  Here ARG1, ARG2 to ARG10 are command line values, which is assigned to corresponding shell variables.    These are also known as special variables provided by the shell. There are some more special variables as given below.   |  |  | | --- | --- | | **Special Variable** | **Variable Details** | | $1 to $n | $1 is the first arguments, $2 is second argument till $n n’th arguments. From 10’th argument, you must need to inclose them in braces like ${10}, ${11} and so on | | $0 | The name of script itself | | $$ | Process id of current shell | | $\* | Values of all the arguments. All arguments are double quoted | | $# | Total number of arguments passed to script | | $@ | Values of all the arguments | | $? | Exit status id of last command | | $! | Process id of last command | |
|  |  |
| **Example** | ### Print total arguments and their values  echo "Total Arguments:" $#  echo "All Arguments values:" $@  ### Command arguments can be accessed as  echo "First->" $1  echo "Second->" $2  # You can also access all arguments in an array and use them in a script.  args=("$@")  echo "First->" ${args[0]}  echo "Second->" ${args[1]}  Now execute this script with 2 arguments and found following results  **Output**  ./arguments.sh Hello ManojKavedia  Total Arguments: 2  All Arguments values: Hello ManojKavedia  First->Hello  Second->ManojKavedia  First->Hello  Second->ManojKavedia |
|  |  |
| **Shell Script** | #!/bin/bash  # EXP ID  # > 7.1  # AIM  # > demo command line args positional and using flags loopsand others  # CODE  echo "Displaying Command line arguments"  echo  echo "1. Positional parameters:"  echo "Arg1: $1"  echo "Arg2: $2"  echo "Arg3: $3"  echo "DONE!!"  echo |
| **Output** |  |
|  |  |
| **Shell Script** | #!/bin/bash  # EXP ID  # > 7.2  # AIM  # > to show positional arguments using flags  # CODE  echo "Displaying Flags"  while getopts "1:2:3:" flag; do  case "${flag}" in  1) argument1=${OPTARG} ;;  2) argument2=${OPTARG} ;;  3) argument3=${OPTARG} ;;  \*) echo "Unknown flag: ${flag}" ;;  esac  done  echo "First: $argument1"  echo "Second: $argument2"  echo "Third: $argument3" |
| **Output** |  |
|  |  |
| **Shell Script** | #!/bin/bash  # EXP ID  # > 7.3  # AIM  # > show arguments using loops  # CODE  echo "3. Using loops"  i=1  for arg in "$@"  do  echo "Argument no. $i: $arg"  i=$(($i+1))  done |
| **Output** |  |
|  |  |
|  |  |
|  |  |
| **Conclusion** | Learned to handle arguments in linux scripts. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Signature** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Date** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Grade** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |