Exit status, command test, flow control.

Jan Trdlička

Department of Computer Systems
Faculty of Information Technology
Czech Technical University in Prague

trdlicka@fit.cvut.cz

October 23, 2018

Contents

Test 1

- Before test
  - Boot the local PC into the Progtest image.
  - Open two terminals.
    - The first terminal for CLI.
    - The second terminal for UNIX manual pages.
- Recommendation
  - Read question carefully.
  - Try to find a correct solution of the question.
  - Verify your solution in CLI.
  - Write down your solution into a paper test.
- Test conditions
  - Work alone!!!
  - Use only UNIX manual pages, no other materials!!!
  - Violations of test conditions means a zero rating.

Exit status

- What is the exit status?
  - The exit status is the value returned by the child process while terminating (exit status falls between 0 and 255).
  - An exit status of zero indicates success.
  - A non-zero exit status indicates failure.
- What is the meaning of exit status of the following commands?
  - ```
    PASS=/etc/passwd
    grep ^root: "$PASS" ; echo $?
    ```

    Pattern is found.
  - ```
    grep ^roooot: "$PASS" ; echo $?
    ```

    Pattern is not found.
  - ```
    grep ^root: /etc/foo ; echo $?
    ```

    Wrong argument.
  - ```
    ~/.bash_history ; echo $?
    ```

    Wrong permissions.

Exit status

- What is the meaning of exit status of the following commands?
  - `winzip ; echo $?`

    Command not found.
  - `ls -lR /      ^C ; echo $?`

    Exit by signal.

Command test and if

- Create a shell script list1.bash that requires one argument.
  - If the argument is missing or if there are more arguments, then the script prints the following error and exits with exit status 1.

```
Usage: ./list1.bash directory_name
```

  - If the argument is a name of a directory and the directory is readable, then the scripts lists its contents by command `ls -la`. Otherwise the script prints the following error and exits with exit status 2.

```
./list1.bash: "foo" is not readable directory
```

- Change the access permissions of the script by the following command

```
chmod 755 list1.bash
```

- Run the script by the following command

```
./list1.bash
```

Command test and if

```
#! /bin/bash

#--- Checking Arguments ---
if [ $# != 1 ]
  then
    echo "Usage: $0 directory_name" >&2
    exit 1
fi

if [ -d "$1" -a -r "$1" ]
  #--- For readable directory
  then
    ls -la "$1"

  #--- Otherwise
  else
    echo "$0: \"$1\" is not readable directory" >&2
    exit 2
fi
```

Loop for

- Modify previous shell script such that for every regular readable file in the directory it will print file type (output from command file).
- Example of script output

```
$> ../list2.bash /usr/bin/ | head

/usr/bin/[: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked
/usr/bin/411toppm: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
/usr/bin/7z: POSIX shell script, ASCII text executable
/usr/bin/7za: POSIX shell script, ASCII text executable
/usr/bin/a2p: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
/usr/bin/aa-easyprof: Python script, ASCII text executable
/usr/bin/abs2rel: Lua script, ASCII text executable
/usr/bin/aclocal: awk script, ASCII text
/usr/bin/aclocal-1.13: awk script, ASCII text
/usr/bin/aconnect: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
```

Loop for

```
...
if [ -d "$1" -a -r "$1" ]
  #--- For readable directory
  then

    for i in "${1%/}"/*     # Remove matching suffix pattern
    do
      if [ -f "$i" -a "$i" ]
        then
          file "$i"
      fi
    done

  #--- Otherwise
  else
    echo "$0: \"$1\" is not readable directory" >&2
    exit 2
fi
```

Homework

- Create a shell script list3.bash that requires one argument.
  - If the argument is missing or if there are more arguments, then the script prints the following error and exits with exit status 1.

```
Usage: ./list3.bash directory_name
```

  - If the argument is a name of a directory and the directory is readable, then the scripts counts the number of subdirectories of this directory and prints this information in the following format:

```
$> ./list3.bash /etc/init.d/rc0.d/
/etc/init.d/rc0.d/: no subdirectory
```

```
$> ./list3.bash /usr/bin/
/usr/bin/: 1 subdirectory
```

```
$> ./list3.bash /etc
/etc: 134 subdirectories
```