NOTES: Bash Scripting

02 May. 2012

Benoit RAT

Contents

1 Special keywords 2

2 Parsing arguments 2

3 Path 3

3.1 Obtain full root dir of the script 3

3.2 Obtain filename 3

4 Checking 3

4.1 return agurment 3

4.2 Bash special checking 3

4.3 OR 3

5 Redirection 4

1 Special keywords

- `!$` Last arguments in shell
- `!*` All last arguments
- `!!` Last command in shell

2 Parsing arguments

Below one of the best way to parse args in bash script

```
#!/bin/bash
echo "OPTIND starts at $OPTIND"
while getopts ":pq:" optname
  do
    case "$optname" in
      "p")
        echo "Option $optname is specified"
        ;;
      "q")
        echo "Option $optname has value $OPTARG"
        ;;
      "?")
        echo "Unknown option $OPTARG"
        ;;
      ":")
        echo "No argument value for option $OPTARG"
        ;;
      *)
      # Should not occur
        echo "Unknown error while processing options"
        ;;
    esac
    echo "OPTIND is now $OPTIND"
  done
```

Otherwise you can try an easyway to parse it withou getopts

```
while [ $# -gt 0 ]; do    # Until you run out of parameters . . .
    case "$1" in
        -h|--help) showhelp;;
        -c|--clean) rm -f $WRS_DONE_DIR/00*;;
        -f|--fetch) echo "To be done";;
        -l|--list) cd $WRS_DONE_DIR; ls 0*; exit 0;;
        --step=0[0-9]) num=$(echo $1 | sed -e 's/--step=//');;
        *) showhelp;;
    esac
    shift       # Check next set of parameters.
done
```

3 Path

3.1 Obtain full root dir of the script

```
root="$(dirname $(/bin/pwd)/$0)"
```

You should use `dirname` or `basename` commands which are easier than sed

3.2 Obtain filename

You can use `basename` function

Or you can find what is before the first dot by doing

```
for f in `ls *.eps`; do
    echo -n "Converting ${f%.*} ... "
    convert -density 100 $f -flatten ${f%.*}.png;
    echo " OK"
done
```

4 Checking

4.1 return agurment

```
which pandoc > /dev/null
if [ $? -ne 0 ]; ## If there is an error in previous command

fi
```

4.2 Bash special checking

```
if [[ -z $var ]]; then ...; fi
```

Where we can use BASH conditional:

- `-z` = empty;
- `-n` = notempty;
- `-f` = file exist;
- `-d` = directory exist

4.3 OR

```
if [[ $1 == "cviugr-v2" || $1 == "RECOMP" ]]; then
    ## Backup WORK folder
    echo "OK"
else
    echo "ERR"
fi
```

5 Redirection

- stdout 2 file: `ls -l > ls-l.txt`
- stderr 2 file: `grep da * 2> grep-errors.txt`
- stdout & stderr 2 file: `rm -f $(find / -name core) &> /dev/null`
- stdout 2 stderr: `grep da * 1>&2`
- stderr 2 stdout: `grep da * 2>&1`