CSC209 Week 2 Notes

Hyungmo Gu

May 10, 2020

Shell Programming 1 of 6

- *.<EXTENSION>
  - returns all items under the extension

```
    # computer-science-notes folder
    >>> echo *.pdf
    common_mistakes_in_proofs.pdf sample.pdf

```

- *read*
  - store values in variable interactively“

```
    >>> read x y
    >>> hello world 2
    >>> echo $x
    hello
    >>> echo $y
    world 2

```

- “
  - store commands in variable

```
    >>> i = ‘expr 4 + 1‘
    >>> echo $i
    5

```

Shell Programming 2 of 6

- *test*
  - checks file types and compares values
  - used in if and while statement to check condition

```
    >>> cat
    >>> echo $?
    0

```

  - **test <EXPRESSION >** is equivalent to [**EXPRESSION**]

```
    >>> if [ 0 -eq 0 ]
    if> then
    then> echo 'hello'
    then> fi
    hello

```

  - has the following numeric comparison operators
    1. **-lt:** less than
    2. **-gt:** greater than
    3. **-eq:** equal to
    4. **-ne:** not equal
    5. **-le:** less than
    6. **-ge:** greater than

```
    >>> test 2 -lt 3
    >>> echo $?
    0

```

  - has the following file testing operators
    1. **-f file:** file exists and is a plain file
    2. **-d file:** file exists and is a directory
    3. **-s file:** file exists and is a plain file of non-zero size

```
    >>> echo 'hello world' > sample.txt
    >>> test -s sample.txt
    >>> echo $?
    0

```

- *if*
  - **&&:** chains multiple if conditions

```
    >>> if [ 0 -eq 0 ] && [ 1 -eq 1 ]
    if> then
    then> echo 'hello'
    then> fi
    hello

```

Shell Programming 3 of 6

- Quoting in Sh
  - Double quotes supress the interpretation of everything except for
    1. Dollar sign
    2. Backquote
    3. Backslash
  - Single quote suppresses interpretation of everything
- Switch
  - '*' acts as a wild card like regex
  - '*' is also used as else

```
    >>> cat switch_example.sh
    array=("hello" "hello there" "goddbye" 2)
    for i in "${array[@]}"
    do
        case $i in
            hello*)
                echo 'hello there'
                ;;
            goodbye)
                echo 'see you later'
                ;;
            *)
                echo "it's a pleasure to meet you"
                ;;
        esac
    done
    >>> sh switch_example.sh
    hello there
    see you later
    it's a pleasure to meet you

```

- *seq*

– works like *range* in python

```
    >>> seq 1 4
    1
    2
    3
    4
    >>> cat seq_example.sh
    for i in `seq 1 4`
    do
        echo $i;
    done
    >>> sh seq_example.sh
    1
    2
    3
    4

```