1 Shell CheatSheet Languages

- PDF Link: cheatsheet-shell-A4.pdf, Category: languages
- Blog URL: `https://cheatsheet.dennyzhang.com/cheatsheet-shell-A4`

File me Issues or star this repo.
See more CheatSheets from Denny: #denny-cheatsheets

1.1 Basic

Name	Comment
Redirect stdout/stderr	`ls /tmp >/dev/null 2>&1`
Deal with filename	`basename $f`, `dirname $f`
Use timeout: avoid command hang	timeout 10 sh -c 'ls -lt'
Restart shell without killing terminal	`exec -l $SHELL`
Run sub-shell	`echo $BASH_SUBSHELL; ( echo "Running in subshell:  $BASH_SUBSHELL" )`
Run static code check	link: shellcheck
Trap exit signal	code/trap-exit.sh
shell retry	code/shell-retry.sh

1.2 GNU tools

1.2.1 Check file

Name	Comment
Show file content	`cat /etc/hosts`
Show file content with line numbers	`cat -n /etc/hosts`
Show with line numbers, while excluding blank lines	`cat -b /etc/hosts`
Show the first 3 lines	`head -n3 /etc/hosts`
Show the last 3 lines	`tail -n3 /etc/hosts`
Keep tailing log files	`tail -f /var/log/system.log`, `taif /var/log/system.log`
Show the 4th line	`sed -n '4p' /etc/hosts`
Show 4th, 5th, 7th and 8th lines	`sed -n '4,5p;7,8p' /etc/hosts`
Show matched string with 3 lines before and after	`grep -C 3 "127.0.0.1" /etc/hosts`
For table-like files, show 2nd column	`awk -F'\t' '{print $2}' /etc/hosts`
For table-like files, swap 1st and 2nd columns	`awk -F'\t' '{print $2,$1}' /etc/hosts`

1.2.2 Copy file

Name	Comment
Copy one file	`cp /etc/hosts /tmp/hosts`
Copy one folder	`cp -r /usr/local/bin/ /tmp/bin/`
Copy for backup	`cp /tmp/hosts{,.bak}`, `ls -lth /tmp/hosts*`
Create a copy but ask confirmation for overwrite	`cp -i ~/Desktop/foo.txt ~/Documents/foo.txt`
Create a copy for backup with timstamp as suffix	`cp foo.txt{,."$(date +%Y%m%d-%H%M%S)"}`

1.2.3 Watch files

Name	Comment
Show file changes	`watch -d -n 1 stat /var/log/message`
Keep tailing log files	`tail -f /var/log/system.log`, `taif /var/log/system.log`

1.2.4 Echo string

Name	Comment
Echo red text	echo -e "hello,[0;31m there [0;31m"
Echo multiple lines	echo -e "hello,"
Echo bold text	echo -e hello, "\033[1mThis is bold text.\033[0m"
Echo underlined text	echo -e hello, "\033[4mThis is underlined text.\033[0m"

1.2.5 Check process via /proc

Name	Comment
Check process start command	`cat /proc/$pid/cmdline`
Check process environment variables	`cat /proc/$pid/environ`
Check process ulimits setting	`cat /proc/$pid/limits`

1.3 Shell Basic

1.3.1 cd

Name	Comment
Go to given folder	`cd /var/log/`
Go to folder in subshell	`(cd /var/log/ && ls)` After this, PWD won’t be changed
Go to home	`cd`
Go to parent folder	`cd ..`
Go to previous folder	`cd -`

1.3.2 Numeric

Name	Comment
*	`expr 5 \* 4`
+	`let z=x+y, z=$x+$y`
==	`int1 -eq int2`, `[ $?  -eq 0 ] && echo "good"`
>=	`int1 -ge =int2`
>	`int1 -gt =int2`
<=	`int1 -le =int2`
<	`int1 -lt =int2`
!=	`int1 -ne =int2`

1.3.3 xargs

```
# Run grep for files filtered by find
find /var/log -name "*.log" | xargs grep -i error

# Loop with pipes
cat /etc/passwd | awk -F’:’ ’{print $1}’ | xargs -I{} sudo -l -U {} | grep -v "not allowed to"
```

1.4 Scripts

- Run command with retry

```
function retry_command {
    local command=${1?}
    local timeout_seconds=${2?}
    local check_interval=${3:-"3"}
    n=0
    until [ "$n" -ge "$timeout_seconds" ]
    do
        if eval "$command" >/dev/null 2>&1; then
            return
        fi
        n=$((n+check_interval))
        echo "Sleep $check_interval seconds: $command"
        sleep "$check_interval"
    done
    echo "After waiting for $timeout_seconds seconds, it still fails"
    exit 1
}

# retry_command "ls /etc/hosts" 6 2
# retry_command "ls /etc/hostss" 6 2
```

- Log with timestamp

```
function log {
    local msg=$*
    date_timestamp=$(date +[’%Y-%m-%d %H:%M:%S’])
    echo -ne "$date_timestamp $msg\n"

    if [ -n "$LOG_FILE" ]; then
        echo -ne "$date_timestamp $msg\n" >> "$LOG_FILE"
    fi
}

# log "hello, world"
```

- Confirm current user is root

```
function ensure_is_root {
    # Make sure only root can run our script
    if [[ $EUID -ne 0 ]]; then
        echo "Error: This script must be run as root." 1>&2
        exit 1
    fi
}
```

- Compare command output

```
[ 0 -eq $(find ./data -name "*.txt" -type f -print | wc -l) ]
```

- get ip from eth0

```
/sbin/ifconfig eth0 | grep ’inet addr:’ | cut -d: -f2 | awk ’{ print $1}’
```

- Check if a string contains a substring

```
#!/bin/bash

S="Pineapple"

if [[ "${S}" == *apple* ]]; then
    echo "Yes"
else
    echo "No"
fi
```

- Check if a string in a list

stackoverflow link

```
#!/usr/bin/env bash
IAAS_TYPE="gcp|aws"
TYPE="gcp"

if [[ "gcp|aws" == *${TYPE}* ]]; then
    echo "yes"
else
    echo "no"
fi
```

1.5 More Resources