

???????

??????

- [Use Bash Strict Mode](#)

```
# quickly syntax
set -euo pipefail

# let script exit if a command fails
set -o errexit
# OR
set -e

# let script exit if an unused variable is used
set -o nounset
# OR
set -u

# This setting prevents errors in a pipeline from being masked.
set -o pipefail

# for Debug
set -x

# setting IFS
IFS=$'\n\t'
```

How `set -o pipefail` works

```
$ grep some-string /non/existent/file | sort
grep: /non/existent/file: No such file or directory
$ echo $?
0

$ set -o pipefail
```

```
$ grep some-string /non/existent/file | sort
grep: /non/existent/file: No such file or directory
$ echo $?
2
```

How the `IFS` works

```
#!/bin/bash
names=(
  "Aaron Maxwell"
  "Wayne Gretzky"
  "David Beckham"
  "Anderson da Silva"
)

echo "With default IFS value..."
for name in ${names[@]}; do
  echo "$name"
done

echo ""
echo "With strict-mode IFS value..."
IFS=$'\n\t'
for name in ${names[@]}; do
  echo "$name"
done

##### Output #####
With default IFS value...
Aaron
Maxwell
Wayne
Gretzky
David
Beckham
Anderson
da
Silva

With strict-mode IFS value...
```

Aaron Maxwell
Wayne Gretzky
David Beckham
Anderson da Silva

???????

```
# [ ] [ ] [ ] [ ]  
readlink -f <file.name>  
  
# [ ] [ ] [ ]  
WORKDIR=$(readlink -f "$0") ;[ ] [ ] [ ] [ ] [ ] [ ] [ ]  
WORKDIR=$( cd $( dirname "$0" ) && pwd )
```

Script ????

```
$ echo $0  
./test.sh  
  
$ echo `basename $0`  
test.sh
```

??????????

```
cat <<EOF  
Welcome .....  
[ ]  
Here are the messages that you want to show up  
[ ]  
EOF
```

??????????????

Sample #1

```
#!/bin/sh  
LOG="my.log"  
(  
....  
) 2>&1 | tee -a $LOG
```

???????? python ??????????????????????

Sample #2

```
temp=$(mktemp)
exec &> ${temp}

echo "All outputs will be saved into the file ${temp}."
```

Sample #3

```
#!/bin/bash
set -eu
exec 3>&1 4>&2
trap 'exec 2>&4 1>&3' 0 1 2 3
exec 1>/path/to/script.log 2>&1

# rest of the script below
dnf -y in foo bar
# firewall rules goes here
....
...
```

??????????????

```
## *.old -> *.new
for fname in $(ls *.old);do echo "mv $fname ->"; echo $(echo $fname |sed 's/.old/.new/');mv $fname $(echo
$fname | sed 's/.old/.new/');done
```

??????????????

```
mkdir {AAA,BBB,CCC}
```

????????

```
cp my.cfg{,.bak}
```

????????

? find

```
# [REDACTED]
```

```
find . -type f | wc -l
```

```
# [REDACTED]
```

```
find . -maxdepth 1 -type d -print0 | xargs -0 -l {} sh -c 'echo $(find {} -type f | wc -l) {}' | sort -n
```

```
NOTE: [REDACTED]
```

? rsync

```
rsync --stats --dry-run -ax /usr /tmp
```

Number of files: 326,373 (reg: 211,698, dir: 24,284, link: 90,391)

Number of created files: 326,373 (reg: 211,698, dir: 24,284, link: 90,391)

Number of deleted files: 0

Number of regular files transferred: 211,698

Total file size: 7,180,685,730 bytes

Total transferred file size: 7,178,524,818 bytes

```
NOTE: [REDACTED]/tmp [REDACTED] reg: 211698 [REDACTED]
```

? tree

```
tree /mydir -a | tail -n 1
```

5 directories, 56 files

```
NOTE: [REDACTED] symbolic link [REDACTED]
```

???? SHELL ???

1. ?? pipe line

```
echo"md5sum $X > $X.sum "| bash
```

2. ?? eval

```
get_arch="uname -p"
```

```
if [ "`eval "$get_arch"`" = "i686" ]; then
```

```
....
```

```
fi
```

???????????

?????

```
PROMPT_DIRTRIM=2
```

?? CSV ?

- [Doing a database join with CSV files](#)
- [tvs-utils](#) - eBay's TSV Utilities: Command line tools for large, tabular data files. Filtering, statistics, sampling, joins and more.
- [How To Parse CSV Files In Bash Scripts In Linux](#)
- [How to convert JSON to CSV using Linux / Unix shell](#)

```
#!/bin/bash
INPUT=data.csv
OLDIFS=$IFS
IFS=' '
[ ! -f $INPUT ] && { echo "$INPUT file not found"; exit 99; }
while read fname dob ssn tel status
do
    echo "Name : $fname"
    echo "DOB : $dob"
    echo "SSN : $ssn"
    echo "Telephone : $tel"
    echo "Status : $status"
done < $INPUT
IFS=$OLDIFS
```

CSV and JSON

```
# JSON to CSV
cat df.json | jq -r '.[0] | join(",")'
cat bingbot.json | jq -r '.prefixes[] | {cidr: .ipv4Prefix, comment: "BingBot"} | join(",")' > bingbot.csv
```

JSON ?

- [GitHub] [gron - Make JSON greppable!](#)

?????

```
tmpfile1=$(mktemp)
tmpfile2="/tmp/$(basename $0).$$tmp"
```

Get my public IP

```
curl ifconfig.me
curl ifconfig.me/ip
curl ifconfig.co
curl checkip.amazonaws.com
curl icanhazip.com
curl ipecho.net/plain

dig +short myip.opendns.com @resolver1.opendns.com
dig TXT +short o-o.myaddr.l.google.com @ns1.google.com
dig TXT +short o-o.myaddr.l.google.com @ns1.google.com | awk -F'"' '{print $2}'
```

?? IP

```
# On Linux
hostname -I
hostip=$(/sbin/ip a | awk '/eth[012]:|ens192:|bond0:|/^$/ ' | grep -E "inet [0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}" | head -1 | awk -F " " '{print $2}' | cut -d"/" -f1)

# On AIX
hostip=$( ifconfig -a | grep inet | awk '{print $2}' | head -1 )
hostip=$( ifconfig -a | grep inet | awk '{print $2}' | sed -n 's/^\([0-9]\{1,3\}\.\)\{3\}[0-9]\{1,3\}/p' | head -1 )
```

??????????

```
# Way 1
du -a /var | sort -n -r | head -n 10

# Way 2
cd /path/to/some/where
du -hsx * | sort -rh | head -10
du -hsx -- * | sort -rh | head -10

# Way 3
find /path/to/dir/ -printf '%s %p\n' | sort -nr | head -10
```

```
find . -printf '%s %p\n' | sort -nr | head -10

## Skip directories and only display files

find /path/to/search/ -type f -printf '%s %p\n' | sort -nr | head -10


# Create a shell alias

## shell alias ##

alias ducks='du -cks * | sort -rn | head'

## deal with special files names ##

alias ducks='du -cks -- * | sort -rn | head'
```

????????

```
# With tr

for f in *; do mv "$f" `echo $f | tr ' ' _`; done


# With find

find . -type f -name "* *.xml" -exec bash -c 'mv "$0" "${0// /_}"' {} \;
```

????

```
echo -n "MySQL username: " ; read username

echo -n "MySQL password: " ; stty -echo ; read password ; stty echo ; echo
```

shuf: ????

```
curl -s https://www.imdb.com/list/ls020046354/export | cut -d ',' -f 6 | shuf
```

tree: ??????????

```
tree --dirsfirst --filelimit 10 --sort=name


# Display size of files

tree -s


# Display permissions of files

tree -p


# Display directory only

tree -d
```



```
# Display till a certain level/depth
```

```
tree -L 1
```

```
# List only those files that match pattern given
```

```
tree -P *screenshot*
```

sort : ?????

```
sort -t ',' -k5,5 -k1,1 -k9,9 -k3,3 -k11,11 my.csv
```

- -t ?????
- -k5,5 ??? 5 ??????????
- ??????????? 5, 1, 9, 3, 11 ?
- ?????????????????? -k5,5n

timeout : ??????

```
timeout 10 tail -f /var/log/httpd/access.log
```

```
timeout 5m ping 8.8.8.8
```

```
timeout 300 tcpdump -n -w data.pcap
```

```
# Sending specific signal
```

```
# To get a list of all available signals, use the command kill -l .
```

```
timeout -s SIGKILL ping 8.8.8.8
```

variables : ??

| ?? | ?? |
|--------|-------------------|
| \$0 | ???? |
| \$1 | ?1??? |
| \$2 | ?2??? |
| \${10} | ?10??? #10 |
| \$# | ????? |
| \$* | ?????? (??????) |
| \$@ | ?????? (????????) |
| \${*} | ???????????? |
| \${@} | ???????????? |
| \$? | ???????? |
| \$\$ | ????? ID |

| | |
|-----------------|------------------------|
| \$_ | ??????????? |
| \$! | ????????? ID |
| u=\${1:-root} | ?? \$1 ???????? root |
| u=\${USER:-foo} | ?? \$USER ???????? foo |
| len=\${#var} | ?? \$var ???? |

?? \$2 ??????????????

```
{varName?Error varName is not defined}
{varName:?Error varName is not defined or is empty}
```

- [Introduction to Bash Shell Parameter Expansions](#)

?????????????

```
mkdir my-dir && cd $_
```

cut: ????

```
# AAA = BBB,  BBB
cut -d= -f2

# 111 2222 33 444444 555,  33 
cut -d ' ' -f3-

# 
head -n 1 data.csv | wc # 
head data.csv | cut -c -30 #  30 ,  30  30-
```

tr: ??

```
# aaa bbb ccc
# 
# aaa
# bbb
# ccc
echo "aaa bbb ccc" | tr " " "\n"
```

xargs

??(Pipe) ??? stdin ???????

- [How to Use the Powerful Xargs Command in Linux](#)

printf: ?????

- %s ??
- %d ??

```
printf "%-40s .....%s\n" "Disable the service $1" "[$2]"
```

```
Disable the service apmd .....[OK]
Disable the service bluetooth .....[OK]
Disable the service hidd .....[OK]
Disable the service cups .....[OK]
Disable the service firstboot .....[OK]
Disable the service readahead_early .....[OK]
```

```
printf "%40s .....%s\n" "Disable the service $1" "[$2]"
```

```
Disable the service apmd .....[OK]
Disable the service bluetooth .....[OK]
Disable the service hidd .....[OK]
Disable the service cups .....[OK]
Disable the service firstboot .....[OK]
Disable the service readahead_early .....[OK]
```

??????

```
printf "%.0f" 9.46666667 # output: 10
```

```
printf "%.2f" 9.46666667 # output: 9.47
```

???????

```
printf -- '-%.0s' {1..80}
```

```
printf -- '=%.0s' {1..80}
```

ping: ????? IP ??

```
{ for p in {1..254}; do ping -c1 -w1 10.22.9.$p & done } | grep "64 bytes"
```

nice: Reduce CPU and Disk load of backup scripts

```
# Reduce the I/O priority of the /usr/local/bin/backup.sh script so that it does not interfere with other processes
```

```
# The -n parameter must be between 0 and 7, where lower numbers mean higher priority
```

```
/usr/bin/ionice -c2 -n7 /usr/local/bin/backup.sh
```

```
# To reduce the CPU priority, use the command nice
```

```
# The -n parameter can range from -20 to 19, where lower numbers mean higher priority
```

```
/usr/bin/nice -n 19 /usr/local/bin/backup.sh
```

```
# Nice and ionice can also be combined, to run a script at low I/O and CPU priority
```

```
/usr/bin/nice -n 19 /usr/bin/ionice -c2 -n7 /usr/local/bin/backup.sh
```

Hex to ASCII

```
# hex = 54657374696e672031203220330
```

```
# ascii = Testing 1 2 3
```

```
# xxd
```

```
echo 54657374696e672031203220330 | xxd -r -p && echo "
```

```
# printf
```

```
printf '\x54\x65\x73\x74\x69\x6e\x67\x20\x31\x20\x32\x20\x33\x0' && echo "
```

```
# sed
```

```
echo -n 54657374696e67203120322033 | sed 's/\([0-9A-F]\{2\}\)/\\\\\\x\1/g' | xargs printf && echo "
```

??????

```
genpasswd() {
```

```
    local l=$1
```

```
    [ "$l" == "" ] && l=16
```

```
    tr -dc A-Za-z0-9_ < /dev/urandom | head -c ${l} | xargs
```

```
}
```

```
tr -dc A-Za-z0-9_ < /dev/urandom | head -c 16 | xargs
```

```
# Generate more than one
tr -dc A-Za-z0-9_ < /dev/urandom | fold -16 | head -5

#
echo FooBar$RANDOM | md5sum | base64 | cut -c 1-12
```

ls: ????

```
# Find the biggest zip file
ls -lSrH ~/Downloads/*.zip
```

stat: ????

```
> stat --printf='Name: %n\nPermissions: %a\n' my.log
Name: my.log
Permissions: 777

> stat --format="%F" my.log
regular file

# Symlink
> stat ~/bin/FoxitReader
File: /home/alang/bin/FoxitReader -> /home/alang/opt/foxitsoftware/foxitreader/FoxitReader.sh
Size: 56      Blocks: 0      IO Block: 4096  symbolic link
Device: 10302h/66306d Inode: 787474   Links: 1
Access: (0777/lrwxrwxrwx)  Uid: ( 1000/  alang)  Gid: ( 1000/  alang)
Access: 2023-07-16 10:26:13.412193581 +0800
Modify: 2023-02-26 12:13:50.234374171 +0800
Change: 2023-02-26 12:13:50.234374171 +0800
Birth: 2023-02-26 12:13:50.234374171 +0800

> stat -L ~/bin/FoxitReader
File: /home/alang/bin/FoxitReader
Size: 120     Blocks: 8      IO Block: 4096  regular file
Device: 10302h/66306d Inode: 1457222   Links: 1
Access: (0755/-rwxr-xr-x)  Uid: ( 1000/  alang)  Gid: ( 1000/  alang)
Access: 2023-07-02 14:34:09.957428285 +0800
Modify: 2023-02-26 12:13:50.246374250 +0800
Change: 2023-02-26 12:13:50.246374250 +0800
Birth: 2023-02-26 12:13:48.530362986 +0800
```

Revision #88

Created 1 June 2020 06:11:50 by Admin

Updated 4 March 2025 15:49:00 by Admin