

Contents

1	small tips	1
1.1	default directory	1
1.2	number list	1
1.3	calculate	2
1.4	lambda function	2
1.5	hex conversion	2
1.6	mail in command line	2
1.7	cron table	2
1.8	merge output	3
1.9	distro check	3
1.10	different authority for different user	3
1.11	use footnote in table or other environments	3
1.12	use locate to find directories	3
1.13	list directory by size	3
1.14	get some informations in sqlite	4
1.14.1	get column names from certain table	4
1.14.2	get tables names from current dbfile	4
1.15	batch rename file with sequential numbers	4
1.16	some tips of regular expression	4
1.16.1	find pattern in many files	5
1.17	change the calling priority of commands	5
1.18	some things for git	5
1.19	reverse string	5

1 small tips

1.1 default directory

the directory of xdg-default were written in `/.config/xdg-dirs.dirs`. For example, if you want to set the default directory gnome search for wallpaper, you can change the value of "XDG_PICTURE_DIR" in that file to the directory path you want to.

1.2 number list

to put a list of continuous of number in vim, use `:put = range(n,m)`, and will generated a list from n to m in document. if want random list, should use shell command `! echo $RANDOM` ¹, and m is the limit of random.

to generate a continuous number list by shell, use `seq $BEGIN $END`, so for loop can use: `for i in $(seq $BEGIN $END);do ...` or `for i in {$BEGIN .. $END}; do ...`

¹:`! echo $((RANDOM%m))` seems can not be used in vim

1.3 calculate

to calculate the sum of the first line in file, use:

```
awk 's+=$1 END print s' file_include_data
```

if the data is not in the first line, change \$1.

and some times awk will get into bug if the sum exceeding 2^{31} , so use printf is a good choice: change the 'print s' to 'printf "%.0f", s'.

1.4 lambda function

python lambda function example: test = lambda x, n: [x[i:i+n] for i in range(0, len(x), n)] this can apart the string every n chars ², and lambda function usually used for the functions needed in a short time, it can also use with filter() or map() function, the syntax of lambda function: "lambda arguments: expression"

1.5 hex conversion

```
awk '{print "ibase=10;obase=2;" $1 }' $file |bc |xargs printf "%08d\n"
```

this can convert every line in \$file from decimal to binary of 8 bits.

"xxd -p" can print only the value, no line number and characters.

python convert hex to bin: bin(int³(str, 16))[2:]

1.6 mail in command line

config before use "mail" command to send e-mail: edit config file⁴, add these lines:

```
set from="your_email_address" smtp="smtp_address_of_your_mail_server"
set smtp-auth-user="your_email_address" smtp-auth-password="your_password"
set smtp-auth=login set smtp-use-starttls(this enable SSL)
```

1.7 cron table

set cron work:

use crontab -e, and add work in the file, the format is: *min hour day month year work*.

²this can use re.findall('..?',str) to replace

³int() function have two arguments, the first is string of number, and the last is the base of this number, for example, int('AB9', 16) can get 2745((2745)₁₀ = (AB9)₁₆).

⁴archlinux use /etc/mail.rc, ubuntu use /etc/s-nail.rc

1.8 merge output

if you want to use the output of multi command as input by pipe, then you need to parenthesis these two commands, such as:

```
(echo "test" ; cat hello.txt) |mail -s "test" username@mailaddress
```

1.9 distro check

if you want to check your distro without screenfetch or neofetch installed, you can use: "cat /etc/*-release", then it will output the distro information of your distro.

1.10 different authority for different user

if you want to make different users have different authority to a certain file, you can use *setfacl*: for example, if you want to set all authority to *file* file with group *group*, you can use:

```
setfacl -m g:group:rwX -R5 file
```

and then use: "setfacl -m g:⁶test:r -R file" to set only read authority to group test.

1.11 use footnote in table or other environments

use \footnotemark in the position you need to note, and then use \footnotetext{*your_footnote*} out the environment.

1.12 use locate to find directories

locate didn't have that option, but we can use the feature of locate to get that:

```
locate -b '/dir_name'
```

or you can use regex:

```
locate -r dir_name$
```

⁵-R is to use recursive

⁶for user, use u:

1.13 list directory by size

```
du -sh -B BLOCKSIZE */ |sort -nr
```

This command will list your subdirectory in current directory by size and from big to small, *BLOCKSIZE* such as M, it will output the size by certain format, */ is path, you can also use another path replace it, but you should add “/*” in the tail.

```
du -sh */ |sort -h
```

This will directly sort by size, not need to appoint block size.

1.14 get some informations in sqlite

1.14.1 get column names from certain table

if you are in sqlite command, you can execute ‘.schema *table_name*’ to get it;
or if you are in python or you just want to use sql command to get it, you can try:

```
SELECT sql FROM sqlite_master WHERE tbl_name = ‘table_name’ AND type = ‘table’
```

1.14.2 get tables names from current dbfile

if you are in sqlite command, you can execute ‘.tables’ to get it;
or if you are in python or you just want to use sql command to get it, you can try:

```
SELECT name FROM sqlite_master WHERE type = ‘table’
```

1.15 batch rename file with sequential numbers

Firstly, set a variable:

```
a=1
```

and then execute the while loop (or for loop):

```
ls *.jpg | while read line;do mv $line ` printf “%03d.jpg” “$a” ` ;let a=a+1;done
```

1.16 some tips of regular expression

To change a certain line a a certain text block, you can use:

```
sed -i "/pattern1/,/pattern2/s/origin_pattern/dest_pattern/g" filename
```

1.16.1 find pattern in many files

```
grep -rnw '/path/to/destination' -e 'pattern'
```

1.17 change the calling priority of commands

To call commands in linux, the system will find the command in the paths which *PATH* variable stores, and the order of path in *PATH* variable depend the priority of calling commands. So, you can adjust the order of paths in *PATH* variable to make you call certain commands.

For example, if you have a different version of gcc from gcc in your system in your current directoy, you can use

```
export PATH=.:$PATH$
```

and then when you type */usr/bin/envgcc*, or execute the program include it, you use the one in your current directory ⁷

1.18 some things for git

1. tracking

- `git add -A` stages All
- `git add .` stages new and modified, without deleted
- `git add -A` stages modified and deleted.

2. about github: if you want to amend or cancel a commit⁸, use `git push -f github_repository branch` to force push to your github repository.

⁷It's temporarily, it malfunctions after you logout the shell, expect you put this in your *.bashrc*.

⁸use `git reset --soft HEAD~1`

1.19 reverse string

- if just reverse the order of lines in file, use `tac filename`
- if you want to reverse the order of every character in file, use `tac -r -s '.' filename`