

운영체제 및 실습 - Shell -

1

한예진, 김수창, 이승준

Dankook University

- Vim Plugin
- Shell Script

- Vim Vundle

- Vundle git 페이지에서 복사하여 다음과 같은 명령어를 수행
- 위치 : home

```
git clone https://github.com/VundleVim/Vundle.vim.git ~/.vim/bundle/Vundle.vim
```

- Vim Vundle

- .vimrc 생성
- 위치 : home
- 옆의 내용을 복사 붙여넣기 필요
(git에서 복사 할 것)

```
set nocompatible          " be iMproved, required
filetype off              " required

" set the runtime path to include Vundle and initialize
set rtp+=~/.vim/bundle/Vundle.vim
call vundle#begin()
" alternatively, pass a path where Vundle should install plugins
"call vundle#begin('~/.vim/bundle')

" let Vundle manage Vundle, required
Plugin 'VundleVim/Vundle.vim'

" The following are examples of different formats supported.
" Keep Plugin commands between vundle#begin/end.
" plugin on GitHub repo
Plugin 'tpope/vim-fugitive'
" plugin from http://vim-scripts.org/vim/scripts.html
" Plugin 'L9'
" Git plugin not hosted on GitHub
Plugin 'git://git.wincent.com/command-t.git'
" git repos on your local machine (i.e. when working on your own plugin)
Plugin 'file:///home/gmarik/path/to/plugin'
" The sparkup vim script is in a subdirectory of this repo called vim.
" Pass the path to set the runtimepath properly.
Plugin 'rstacruz/sparkup', {'rtp': 'vim/'}
" Install L9 and avoid a Naming conflict if you've already installed a
" different version somewhere else.
Plugin 'ascenator/L9', {'name': 'newL9'}

" All of your Plugins must be added before the following line
call vundle#end()          " required
filetype plugin indent on  " required
" To ignore plugin indent changes, instead use:
"filetype plugin on
"
" Brief help
" :PluginList    - lists configured plugins
" :PluginInstall - installs plugins; append `!' to update or just :PluginUpdate
" :PluginSearch foo - searches for foo; append `!' to refresh local cache
" :PluginClean   - confirms removal of unused plugins; append `!' to auto-approve removal
"
" see :h vundle for more details or wiki for FAQ
" Put your non-Plugin stuff after this line
```

-
- ```

VIM - Vi IMproved

version 8.1.3741
by Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org

Vim is installing plugins to /home/choi_gunhee/.vim/

bundle
Plugin 'VundleVim/Vundle.vim'
Plugin 'tpope/vim-fugitive'
Plugin 'git://git.wincent.com/command-t.git'
! Plugin 'file:///home/gmarik/path/to/plugin'
Plugin 'rstacruz/sparkup'
Plugin 'scrooloose/nerdtree'
Plugin 'frazrepo/vim-rainbow'
Plugin 'nathanaelkane/vim-indent-guides'
* Helptags

[Done] Installer [Preview][R0] 11,0-1 All [No Name] 0,0-1

```

- .vimrc
  - 위치 : home
  - .vimrc 필요한 plugin 및 옵션 설정

```
drwxrwxr-x 4 choi_gunhee choi_gunhee
-rw----- 1 choi_gunhee choi_gunhee
-rw-rw-r-- 1 choi_gunhee choi_gunhee
drwxrwxr-x 5 choi_gunhee choi_gunhee
-rw-rw-r-- 1 choi_gunhee choi_gunhee
choi_gunhee@gunhee-linux-94:~$
```

```
4096 9월 11 00:56 .vim
13800 9월 12 09:43 .viminfo
1980 9월 11 01:22 .vimrc
4096 8월 29 14:02 .vscode-server
270 3월 13 2022 .wget-hsts
```

- .vimrc
  - 위치 : home
  - .vimrc 필요한 plugin 및 옵션 설정

```
42 " see :h vundle for more details or wiki for FAQ
43 " Put your non-Plugin stuff after this line
```

- .vimrc 파일 아래에는 위와 같은 문구 존재
- 이 문구 아래에 작성





- Colors schem : jellybeans

<https://github.com/nanotech/jellybeans.vim>

- mkdir -p ~/.vim/colors
- cd ~/.vim/colors
- curl -O  
<https://raw.githubusercontent.com/nanotech/jellybeans.vim/master/colors/jellybeans.vim>
- 위 명령 실행

- Colors schem : jellybeans

<https://github.com/nanotech/jellybeans.vim>

colorscheme jellybeans

syntax on

```
45 colorscheme jellybeans
46 syntax on
```

위 옵션을 .vimrc에 입력

```
1 #include <stdio.h>
2
3 int main() {
4 int total = 0;
5
6 printf("hello, world\n");
7 for(int i=0; i<10; i++) {
8 total += i;
9 }
10
11 return 0;
12 }
```

```
1 #include <stdio.h>
2
3 int main() {
4 int total = 0;
5
6 printf("hello, world\n");
7 for(int i=0; i<10; i++) {
8 total += i;
9 }
10
11 return 0;
12 }
```

- Rainbow
  - Plugin 'frazrepo/vim-rainbow'
  - let g:rainbow\_active = 1

```
51 Plugin 'frazrepo/vim-rainbow'
52 let g:rainbow_active = 1
```

<https://github.com/frazrepo/vim-rainbow>

```
rustc::usage(os::args()[0].clone())
}
fn find_cmd(command_string: &str) -> Option<Command> {
 do COMMANDS.iter().find |command| {
 command.cmd == command_string
 }.map_move(|x| *x)
}
fn cmd_help(args: &[-str]) -> ValidUsage {
 fn print_usage(command_string: &str) -> ValidUsage {
 match find_cmd(command_string) {
 Some(command) => {
 match command.action {
 CallMain(prog, _) => println!(
 "The %s command is an alias for the %s program.",
 command.cmd, prog),
 _ => ()
 }
 match command.usage_full {
 UsgStr(msg) => println!("%s\n", msg),
 UsgCall(f) => f(),
 }
 Valid(0)
 },
 None => Invalid
 }
 }
 match args {
 [ref command_string] => print_usage((*command_string).clone()),
 _ => Invalid
 }
}
fn cmd_test(args: &[-str]) -> ValidUsage {
 match args {
 [ref filename] => {
 let p = Path(*filename);
 let test_exec = p.filestem().unwrap() + "test-";
 invoke("rustc", &["--test", filename.to_owned(),
 ~"-o", test_exec.to_owned()], rustc::main_args);
 }
 }
}
```

- Indent-guides
  - Plugin 'nathanaelkane/vim-indent-guides'
  - let g:indent\_guides\_enable\_on\_vim\_startup = 1

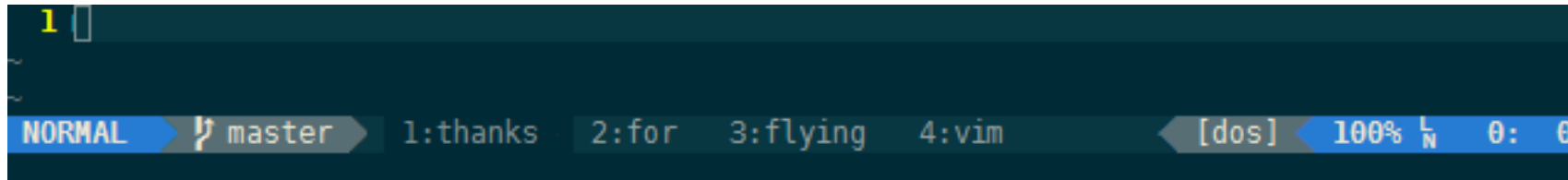
```
54 Plugin 'nathanaelkane/vim-indent-guides'
55 let g:indent_guides_enable_on_vim_startup = 1
```

<https://github.com/nathanaelkane/vim-indent-guides>

```
40 # Enable user to choose a new password
41 def lost_password
42 redirect_to(home_url) && return unless Setting.lost_pass
43 if params[:token]
44 @token = Token.find_by_action_and_value("recovery", pa
45 redirect_to(home_url) && return unless @token and !@to
46 @user = @token.user
47 if request.post?
48 @user.password, @user.password_confirmation = params
49 if @user.save
50 @token.destroy
51 flash[:notice] = l(:notice_account_password_update
52 redirect_to :action => 'login'
53 return
54 end
55 end
56 render :template => "account/password_recovery"
57 return
58 else
59 if request.post?
60 user = User.find_by_mail(params[:mail])
61 # user not found in db
62 end
63 end
64 end
65
66 colorscheme bclear
67 set ts=2 sw=2 et
68 let g:indent_guides_start_level = 2
```

- VIM-AIRLINE
  - Plugin 'vim-airline/vim-airline'

<https://github.com/vim-airline/vim-airline>



- Vim 옵션
  - syntax on : 문법 강조
  - set hlsearch : 검색 매칭 문자열 강조
  - set nu : 줄 번호 표시
  - set autoindent : 자동 들여쓰기
  - set cindent : 자동 들여쓰기 (c언어용)
  - set ts=4 : '/t'에서 몇 개의 스페이스로 할 것인지
  - set sts=4 : tab을 눌렀을 때, 몇 개의 스페이스로 할 것인지
  - set shiftwidth=4 : 자동 들여쓰기 할 때, 몇개의 스페이스로 할 것 인지

- Vim 옵션
  - set laststatus=2 : statusline을 보여주는 값  
(0 :출력안함, 1:창 2개이상 일때, 2:항상)
  - set showmatch : 닫는 괄호 입력 시, 짝이 되는 괄호를 하이라이트
  - set ruler : 커서 위치의 줄번호와 행번호 출력
  - set fileencodings=utf8,euc-kr : 인코딩 형식 지정



- Vim 옵션
  - Smart setting : 검색, 탭, 들여쓰기를 자동으로 설정
    - set smartcase
    - set smarttab
    - set smartindent
  - set ignorecase : 검색 시 대소문자 무시
  - set incsearch : 단어 검색시 글자 입력할 때마다 검색

- Vim 창 분할
  - sp : 가로 창 분할
  - vp : 세로 창 분할
- 이동
  - Crtl + w, w : 다음 창으로 커서
  - Crtl + w, W : 이전 창으로 커서 이동
  - Crtl + w, [H, J, K, L] : 해당 방향으로 커서 이동, vim 에서는 H, J, K, L이 방향키를 대체 가능
- Crtl + w, r : 다음 화면과 위치 바꾸기
- 종료 : q, 모든 창 종료 : qa

- ctags
  - 소스 파일 내에 정의된 전역 변수, 함수, 매크로, 구조체의 정보를 데이터베이스 파일로 생성하는 툴
- 설치
  - `yum install ctags`
- 데이터베이스 생성
  - `ctags -R`
- 경로 설정
  - 위치 : `vimrc`
  - `set tags=/home/user/workspace/project/tags`
  - `set tags=./tags, /usr/src/linux/tags -> 콤마로 여러 개 가능`

- ctags

- 소스 파일 내에 정의된 전역 변수, 함수, 매크로, 구조체의 정보를 데이터베이스 파일로 생성하는 툴

```
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$ ls
arch CREDITS Documentation init kernel Makefile samples tools
block crypto drivers ipc lib mm scripts usr
certs cscope.files fs Kbuild LICENSES net security virt
COPYING cscope.out include Kconfig MAINTAINERS README sound
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$ ctags -R
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$ ls
arch CREDITS Documentation init kernel Makefile samples tags
block crypto drivers ipc lib mm scripts tools
certs cscope.files fs Kbuild LICENSES net security usr
COPYING cscope.out include Kconfig MAINTAINERS README sound virt
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$
```

- ctags
  - 소스 파일 내에 정의된 전역 변수, 함수, 매크로, 구조체의 정보를 데이터베이스 파일로 생성하는 툴

```
set tags=/home/choi_gunhee/MyProject/Linux_Kernel_study/linux-5.16.13/tags
```

- set tags=경로/tags
- 위 옵션을 .vimrc에 입력

- ctags
  - 소스 파일 내에 정의된 전역 변수, 함수, 매크로, 구조체의 정보를 데이터베이스 파일로 생성하는 툴
  - Ctrl + ] : 함수나 구조체 정의로 이동
  - Ctrl + t : 이전 위치로 이동

- cscope
  - ctags로 찾기 힘든 전역 변수나 함수의 호출 등을 볼 때 사용
- 설치
  - `yum install cscope`
- 쉘 스크립트 생성 (mkcscope.sh)
  - `#!/bin/bash`
  - `rm -rf cscope.files cscope.out`
  - `find . `pwd` \$( -name '*.c' -o -name '*.cpp' -o -name '*.cc' -o -name '*.h' -o -name '*.s' -o -name '*.S' \& ) -print > cscope.files cscope -i cscope.files`

- cscope
  - ctags로 찾기 힘든 전역 변수나 함수의 호출 등을 볼 때 사용

- chmod 755 mkcscope.sh
- mv mkcscope.sh /usr/local/bin
  - 언제나 실행하기 위해서 bin으로 이동

```
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$ ls
arch CREDITS Documentation init kernel Makefile samples tags
block crypto drivers ipc lib mm scripts tools
certs cscope.files fs Kbuild LICENSES net security usr
COPYING cscope.out include Kconfig MAINTAINERS README sound virt
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$
```

- 프로젝트 최상위에서 스크립트 실행
  - mkcscope.sh cscope.files cscope.out
  - 생성이 완료되면, Ctrl + d 입력 후 종료



- cscope
  - ctags로 찾기 힘든 전역 변수나 함수의 호출 등을 볼 때 사용
- vimrc에 다음과 같은 내용 입력
  - set csprg=/usr/bin/cscope
  - set cst=0
  - set cst
  - set nocsverb
  - if filereadable("./cscope.out")
  - cs add cscope.out
  - else
  - cs add /usr/user/workspace/project/cscope.out (프로젝트경로)
  - endif
  - set csverb

- cscope
  - ctags로 찾기 힘든 전역 변수나 함수의 호출 등을 볼 때 사용

- 사용법

- cs find [유형] 검색어
  - Ex) cs find 1 inode

|        |                       |
|--------|-----------------------|
| 0 또는 s | C 심볼 검색               |
| 1 또는 g | 정의 검색                 |
| 2 또는 d | 호출되는 함수 검색            |
| 3 또는 c | 호출하는 함수 검색            |
| 4 또는 t | 텍스트 문자열 검색            |
| 6 또는 e | 확장 정규식을 이용한 검색        |
| 7 또는 f | 파일 이름 검색              |
| 8 또는 i | 본 파일을 include하는 파일 검색 |

- 셸 스크립트.
  - Unix, Linux 등에서 사용하는 일반적인 명령어나 if나 for와 같은 프로그래밍적인 요소로 이루어진 인터프리터 기반의 스크립트 언어



- 기초 문법

- 쉘 스크립트는 맨 윗 줄에 `#!/bin/bash`를 붙인다.

- `echo` : 문자열을 컴퓨터 터미널에 출력 명령어

- `vim myshell.sh`

`#!/bin/bash`

`echo "hello, world"`

- `chmod +x myshell.sh`

- `./myshell.sh`

---

```
choi_gunhee@gunhee-linux-94:~/test$ vim myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash
```

```
echo "hello, world"
choi_gunhee@gunhee-linux-94:~/test$ chmod +x myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ ls
myshell.sh test2.txt test.c
choi_gunhee@gunhee-linux-94:~/test$
```

```
choi_gunhee@gunhee-linux-94:~/test$./myshell.sh
hello, world
choi_gunhee@gunhee-linux-94:~/test$
```

- 변수 사용
  - 변수 선언 : language = “Korean”
  - 변수 사용 : \$Korean

```
#!/bin/bash
language="Korean"
echo "hello, $language"
```

```
choi_gunhee@gunhee-linux-94:~/test$ vim myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash
language="Korean"
echo "hello, $language"

choi_gunhee@gunhee-linux-94:~/test$./myshell.sh
"hello, Korean"
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 변수 사용
  - 디렉토리 생성 예제

```
choi_gunhee@gunhee-linux-94:~/test$ ls
myshell.sh test2.txt test.c
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash

name="kor1 kor2"

mkdir $name
choi_gunhee@gunhee-linux-94:~/test$./myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ ls
kor1 kor2 myshell.sh test2.txt test.c
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 함수
  - function 키워드 이용

```
choi_gunhee@gunhee-linux-94:~/test$ vim myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash
```

```
function print() {
 echo "hello, function"
 echo $1
}
```

```
print "hello, $1 print"
choi_gunhee@gunhee-linux-94:~/test$./myshell.sh
hello, function
hello, print
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 전역 변수, 지역 변수
  - Local 키워드 이용

```
choi_gunhee@gunhee-linux-94:~/test$ vim myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash

str1="Hello_str1"

function print() {
 local str2="I am str2"
 echo $str1
 echo $str2
}

print
echo "global $str1"
echo "local $str2"
choi_gunhee@gunhee-linux-94:~/test$./myshell.sh
Hello_str1
I am str2
global Hello_str1
local
choi_gunhee@gunhee-linux-94:~/test$ █
```



- 예약 변수, 환경 변수
  - 시스템에서 미리 정해둔 변수들이 존재
- HOME : 사용자의 홈 디렉토리
- PATH : 실행 파일을 찾을 디렉토리 경로
- PWD : 현재 작업 중인 디렉토리 경로
- USER : 사용자 이름
- OSTYPE : 운영체제 종류

```
choi_gunhee@gunhee-linux-94:~/test$ cat system_v.sh
#!/bin/bash
```

```
echo "Home : $HOME"
echo "Path : $PATH"
echo "pwd : $PWD"
echo "user : $USER"
echo "OS type : $OSTYPE"
```

```
choi_gunhee@gunhee-linux-94:~/test$./system_v.sh
```

```
Home : /home/choi_gunhee
Path : /home/choi_gunhee/anaconda3/bin:/home/choi_gunhee/anaconda3/condabin:/home/choi_gunhee/anaconda3/condabin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
pwd : /home/choi_gunhee/test
user : choi_gunhee
OS type : linux-gnu
choi_gunhee@gunhee-linux-94:~/test$
```

- 매개 변수
  - \$1, \$2, \$3 ... \${10}

```
choi_gunhee@gunhee-linux-94:~/test$ ls
kor1 kor2 myshell.sh system_v.sh test2.txt test.c
choi_gunhee@gunhee-linux-94:~/test$ cat system_v.sh
#!/bin/bash
```

```
echo $1
echo $2
```

```
choi_gunhee@gunhee-linux-94:~/test$./system_v.sh kor1 kor2
kor1
kor2
choi_gunhee@gunhee-linux-94:~/test$ █
```

- \${변수:=문자열}
  - 변수가 설정되어있지 않거나 NULL이라면 문자열로 치환

```
choi_gunhee@gunhee-linux-94:~/test$ cat str_v.sh
#!/bin/bash

var1="kor1"
var2=""

echo "print var1 : $var1"
echo "print var2 : $var2"

echo ""
echo "print var1 : ${var1:=test2}"
echo "print var2 : ${var2:=test2}"
choi_gunhee@gunhee-linux-94:~/test$./str_v.sh
print var1 : kor1
print var2 :

print var1 : kor1
print var2 : test2
choi_gunhee@gunhee-linux-94:~/test$ █
```

- \${변수:?에러 메시지}
  - 변수가 설정되어있지 않거나 NULL이라면  
에러 메시지 출력 후 종료

```
choi_gunhee@gunhee-linux-94:~/test$ cat str_v.sh
#!/bin/bash
```

```
str1="str1"
str2=""
error_msg="Sorry"
```

```
echo ${str1:?$error_msg}
echo ${str2:?$error_msg}
```

```
choi_gunhee@gunhee-linux-94:~/test$./str_v.sh
str1
./str_v.sh: line 8: str2: Sorry
choi_gunhee@gunhee-linux-94:~/test$ █
```

- `${변수:=문자열}`
  - 그 외 변환

| 변환                          | 설명                                      |
|-----------------------------|-----------------------------------------|
| <code>\${변수-문자열}</code>     | 변수가 설정되지 않은 경우, 문자열을 변수로 치환             |
| <code>\${변수:-문자열}</code>    | 변수가 설정되지 않았거나 NULL인 경우 문자열을 변수로 치환      |
| <code>\${변수=문자열}</code>     | 변수가 설정되지 않은 경우, 변수에 문자열을 저장하고 치환        |
| <code>\${변수:=문자열}</code>    | 변수가 설정되지 않았거나 NULL인 경우 문자열을 변수에 저장하고 치환 |
| <code>\${변수:?에러 메시지}</code> | 변수가 설정되지 않았거나 NULL인 경우, 에러메세지 출력하고 종료   |
| <code>\${변수:시작위치}</code>    | 변수값이 문자열인 경우, 시작 위치부터 문자열 길이 끝까지 출력     |
| <code>\${변수:시작위치:길이}</code> | 변수값이 문자열인 경우, 시작 위치부터 길이까지 출력           |

- 문자열 패턴, 변경

| 패턴, 변경              | 설명                                                             |
|---------------------|----------------------------------------------------------------|
| \${변수#패턴}           | 변수 <b>앞</b> 에서부터 <b>처음</b> 찾은 패턴과 일치하는 패턴 <b>앞의 모든</b> 문자열 제거  |
| \${변수##패턴}          | 변수 <b>앞</b> 에서부터 <b>마지막</b> 찾은 패턴과 일치하는 패턴 <b>앞의 모든</b> 문자열 제거 |
| \${변수%패턴}           | 변수 <b>뒤</b> 에서부터 <b>처음</b> 찾은 패턴과 일치하는 패턴 <b>뒤의 모든</b> 문자열 제거  |
| \${변수%%패턴}          | 변수 <b>뒤</b> 에서부터 <b>마지막</b> 찾은 패턴과 일치하는 패턴 <b>뒤의 모든</b> 문자열 제거 |
| \${#변수}             | 변수의 길이                                                         |
| \${변수/찾는문자열/바꿀문자열}  | 변수에서 찾는 문자열을 바꿀 문자열로 변경, 없으면 삭제                                |
| \${변수/#찾을문자열/바꿀문자열} | 변수에서 문자열에서 찾을 문자열 <b>시작과</b> 맞으면 문자열 변경                        |
| \${변수/%찾을문자열/바꿀문자열} | 변수에서 문자열에서 찾을 문자열 <b>마지막과</b> 맞으면 문자열 변경                       |

- 문자열 패턴, 변경

```
choi_gunhee@gunhee-linux-94:~/test$ cat str_change.sh
#!/bin/bash

str1="/etc/test1/hello.txt"

echo $str1
echo ${str1%/*}
echo ${str1##*/}
echo ${#str1}
choi_gunhee@gunhee-linux-94:~/test$./str_change.sh
/etc/test1/hello.txt
/etc/test1
hello.txt
20
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 문자열 패턴, 변경

```
choi_gunhee@gunhee-linux-94:~/test$ cat str_change.sh
#!/bin/bash
```

```
str1="hello choi gunhee hello choi"
```

```
echo ${str1/hello/hi}
echo ${str1//hello/hi}
echo ${str1/hello}
echo ${str1//hello}
echo ${str1/#he/what?}
echo ${str1/%lo/what??}
```

```
choi_gunhee@gunhee-linux-94:~/test$./str_change.sh
```

```
hi choi gunhee hello choi
hi choi gunhee hi choi
choi gunhee hello choi
choi gunhee choi
what?llo choi gunhee hello choi
hello choi gunhee hello choi
```

```
choi_gunhee@gunhee-linux-94:~/test$ █
```



- 조건문

```
If [첫 번째 조건식]
then
 수행문
elif [두 번째 조건식]
then
 수행문
else
 수행문
fi
```

```
choi_gunhee@gunhee-linux-94:~/test$ cat if_case.sh
#!/bin/bash
v1=10
v2=10

if [$v1 = $v2]
then
 echo True
else
 echo False
fi
choi_gunhee@gunhee-linux-94:~/test$./if_case.sh
True
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 조건문

```
case $변수 in
 조건값 1)
 수행문 ;;
 조건값 2)
 수행문 ;;
 *)
 수행문
esac
```

```
choi_gunhee@gunhee-linux-94:~/test$ vim if_case.sh
choi_gunhee@gunhee-linux-94:~/test$ cat if_case.sh
#!/bin/bash
```

```
case $1 in
 hi)
 echo hi
 ;;
 hello)
 echo hello
 ;;
 *)
```

```
esac
```

```
choi_gunhee@gunhee-linux-94:~/test$./if_case.sh hi
hi
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 반복문

```
for 변수 in [범위]
do
 반복할 수행문
done
```

```
choi_gunhee@gunhee-linux-94:~/test$ cat iter.sh
#!/bin/bash

for num in 1 2 3 4 5
do
 echo $num;
done
choi_gunhee@gunhee-linux-94:~/test$./iter.sh
1
2
3
4
5
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 반복문
  - for 변수 in [범위]
  - do
  - 반복할 수행문
  - done

```
choi_gunhee@gunhee-linux-94:~/test$ cat iter.sh
#!/bin/bash

for file in $HOME/*
do
 echo $file;
done
choi_gunhee@gunhee-linux-94:~/test$./iter.sh
/home/choi_gunhee/anaconda3
/home/choi_gunhee/Anaconda3-2021.11-Linux-x86_64.sh
/home/choi_gunhee/cache-trace
/home/choi_gunhee/cscope.files
/home/choi_gunhee/cscope.out
/home/choi_gunhee/data
/home/choi_gunhee/Desktop
/home/choi_gunhee/Documents
/home/choi_gunhee/Downloads
/home/choi_gunhee/Easy-Shell-Script
```

- 반복문

```
for 변수 in [범위]
do
 반복할 수행문
done
```

```
choi_gunhee@gunhee-linux-94:~/test$ cat ./iter.sh
#!/bin/bash
```

```
for num in {1..10..2}
do
 echo $num;
done
```

```
choi_gunhee@gunhee-linux-94:~/test$./iter.sh
```

```
1
3
5
7
9
```

```
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 반복문
  - for 변수 in [범위]
  - do
  - 반복할 수행문
  - done

```
choi_gunhee@gunhee-linux-94:~/test$ cat iter.sh
#!/bin/bash

arr=("a" "b" "c" "d" "e")

for str in ${arr[@]}
do
 echo $str;
done
choi_gunhee@gunhee-linux-94:~/test$./iter.sh
a
b
c
d
e
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 반복문

```
while [$변수1 연산자 $변수2]
do
 수행문
done
```

```
choi_gunhee@gunhee-linux-94:~/test$ cat iter.sh
#!/bin/bash
```

```
n=1
```

```
while [$n -lt 5]
do
```

```
 echo $n
 n=$((n+1))
```

```
done
```

```
choi_gunhee@gunhee-linux-94:~/test$./iter.sh
```

```
1
```

```
2
```

```
3
```

```
4
```

```
choi_gunhee@gunhee-linux-94:~/test$ █
```

- 연산자

| 연산자                    | 설명                                            |
|------------------------|-----------------------------------------------|
| if [ -z \$변수 ]         | 문자열 길이가 0이면 참                                 |
| if [ -n \$변수 ]         | 문자열 길이가 0아니면 참                                |
| if [ \$변수1 -eq \$변수2 ] | 변수1과 변수2의 값이 같으면 참                            |
| if [ \$변수1 -ne \$변수2 ] | 변수1과 변수2의 값이 다르면 참                            |
| if [ \$변수1 -gt \$변수2 ] | 변수1이 변수2보다 크면 참                               |
| if [ \$변수1 -ge \$변수2 ] | 변수1이 변수2보다 크거나 같으면 참                          |
| if [ \$변수1 -lt \$변수2 ] | 변수1이 변수2보다 작으면 참                              |
| if [ \$변수1 -le \$변수2 ] | 변수1이 변수2보다 작거나 같으면 참                          |
| >, >=, <, <=           | 사용시, if ((\$변수 1 > \$변수2))처럼, ()로 한번 더 감싸야한다. |



- grep
  - 특정 디렉토리, 로그 등에서 특정 문자열을 찾는 명령어

- grep [옵션] 패턴 파일

```
choi_gunhee@gunhee-linux-94:~/test$ cat list.txt
total 3219264
drwxrwxr-x 28 choi_gunhee choi_gunhee 4096 5월 18 17:42 anaconda3
-rw-rw-r-- 1 choi_gunhee choi_gunhee 608680744 11월 18 2021 Anaconda3-2021.11-Linux-x86_64.sh
drwxrwxr-x 5 choi_gunhee choi_gunhee 4096 8월 19 11:19 cache-trace
-rw-rw-r-- 1 choi_gunhee choi_gunhee 18930510 3월 24 00:36 cscope.files
-rw-rw-r-- 1 choi_gunhee choi_gunhee 2667770462 3월 24 00:37 cscope.out
drwxrwxr-x 2 choi_gunhee choi_gunhee 4096 8월 19 22:57 data
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Desktop
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Documents
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Downloads
drwxrwxr-x 13 choi_gunhee choi_gunhee 4096 9월 12 19:04 Easy-Shell-Script
drwxrwxr-x 3 choi_gunhee choi_gunhee 4096 5월 18 17:40 kangaroo_project
drwxrwxr-x 7 choi_gunhee choi_gunhee 4096 8월 24 15:39 memcached-1.6.16
-rw-rw-r-- 1 choi_gunhee choi_gunhee 1054877 8월 4 12:36 memcached-1.6.16.tar.gz
drwxrwxr-x 8 choi_gunhee choi_gunhee 4096 6월 17 10:16 MyProject
-rw----- 1 choi_gunhee choi_gunhee 989 6월 7 15:41 nohup.out
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Pictures
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Public
drwxr-xr-x 3 choi_gunhee choi_gunhee 4096 3월 8 2022 snap
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Templates
drwxrwxr-x 4 choi_gunhee choi_gunhee 4096 9월 12 21:49 test
drwxr-xr-x 3 choi_gunhee choi_gunhee 4096 6월 7 19:32 u3_ctrl
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Videos
choi_gunhee@gunhee-linux-94:~/test$ grep -i cscope list.txt
-rw-rw-r-- 1 choi_gunhee choi_gunhee 18930510 3월 24 00:36 cscope.files
-rw-rw-r-- 1 choi_gunhee choi_gunhee 2667770462 3월 24 00:37 cscope.out
choi_gunhee@gunhee-linux-94:~/test$ █
```

- grep
  - 특정 디렉토리, 로그 등에서 특정 문자열을 찾는 명령어

- grep [옵션] 패턴 파일

```
choi_gunhee@gunhee-linux-94:~/test$ cat list.txt
total 3219264
drwxrwxr-x 28 choi_gunhee choi_gunhee 4096 5월 18 17:42 anaconda3
-rw-rw-r-- 1 choi_gunhee choi_gunhee 608680744 11월 18 2021 Anaconda3-2021.11-Linux-x86_64.sh
drwxrwxr-x 5 choi_gunhee choi_gunhee 4096 8월 19 11:19 cache-trace
-rw-rw-r-- 1 choi_gunhee choi_gunhee 18930510 3월 24 00:36 cscope.files
-rw-rw-r-- 1 choi_gunhee choi_gunhee 2667770462 3월 24 00:37 cscope.out
drwxrwxr-x 2 choi_gunhee choi_gunhee 4096 8월 19 22:57 data
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Desktop
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Documents
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Downloads
drwxrwxr-x 13 choi_gunhee choi_gunhee 4096 9월 12 19:04 Easy-Shell-Script
drwxrwxr-x 3 choi_gunhee choi_gunhee 4096 5월 18 17:40 kangaroo_project
drwxrwxr-x 7 choi_gunhee choi_gunhee 4096 8월 24 15:39 memcached-1.6.16
-rw-rw-r-- 1 choi_gunhee choi_gunhee 1054877 8월 4 12:36 memcached-1.6.16.tar.gz
drwxrwxr-x 8 choi_gunhee choi_gunhee 4096 6월 17 10:16 MyProject
-rw----- 1 choi_gunhee choi_gunhee 989 6월 7 15:41 nohup.out
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Pictures
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Public
drwxr-xr-x 3 choi_gunhee choi_gunhee 4096 3월 8 2022 snap
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Templates
drwxrwxr-x 4 choi_gunhee choi_gunhee 4096 9월 12 21:49 test
drwxr-xr-x 3 choi_gunhee choi_gunhee 4096 6월 7 19:32 u3_ctrl
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Videos
choi_gunhee@gunhee-linux-94:~/test$ grep -i cscope list.txt
-rw-rw-r-- 1 choi_gunhee choi_gunhee 18930510 3월 24 00:36 cscope.files
-rw-rw-r-- 1 choi_gunhee choi_gunhee 2667770462 3월 24 00:37 cscope.out
choi_gunhee@gunhee-linux-94:~/test$ █
```

- find
  - 파일을 찾아주는 명령어
  - find [옵션] [경로] [표현식]

```
choi_gunhee@gunhee-linux-94:~/test$ ls
if_case.sh kor1 list.txt msg str_change.sh system_v.sh test.c
iter.sh kor2 log.sh myshell.sh str_v.sh test2.txt
choi_gunhee@gunhee-linux-94:~/test$ find ./ -name list.txt
./list.txt
choi_gunhee@gunhee-linux-94:~/test$ find ./ -name list
choi_gunhee@gunhee-linux-94:~/test$ find ./ -name list*
./list.txt
choi_gunhee@gunhee-linux-94:~/test$ █
```

- awk
  - 특정 인덱스 문자열 출력
  - awk [옵션] 패턴(액션) [대상파일]

```
choi_gunhee@gunhee-linux-94:~/test$ cat list.txt
total 3219264
drwxrwxr-x 28 choi_gunhee choi_gunhee 4096 5월 18 17:42 anaconda3
-rw-rw-r-- 1 choi_gunhee choi_gunhee 608680744 11월 18 2021 Anaconda3-2021.11-Linux-x86_64.sh
drwxrwxr-x 5 choi_gunhee choi_gunhee 4096 8월 19 11:19 cache-trace
-rw-rw-r-- 1 choi_gunhee choi_gunhee 18930510 3월 24 00:36 cscope.files
-rw-rw-r-- 1 choi_gunhee choi_gunhee 2667770462 3월 24 00:37 cscope.out
drwxrwxr-x 2 choi_gunhee choi_gunhee 4096 8월 19 22:57 data
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Desktop
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Documents
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Downloads
drwxrwxr-x 13 choi_gunhee choi_gunhee 4096 9월 12 19:04 Easy-Shell-Script
drwxrwxr-x 3 choi_gunhee choi_gunhee 4096 5월 18 17:40 kangaroo_project
drwxrwxr-x 7 choi_gunhee choi_gunhee 4096 8월 24 15:39 memcached-1.6.16
-rw-rw-r-- 1 choi_gunhee choi_gunhee 1054877 8월 4 12:36 memcached-1.6.16.tar.gz
drwxrwxr-x 8 choi_gunhee choi_gunhee 4096 6월 17 10:16 MyProject
-rw-rw-r-- 1 choi_gunhee choi_gunhee 989 6월 7 15:41 nohup.out
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Pictures
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Public
drwxr-xr-x 3 choi_gunhee choi_gunhee 4096 3월 8 2022 snap
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Templates
drwxrwxr-x 4 choi_gunhee choi_gunhee 4096 9월 12 21:49 test
drwxr-xr-x 3 choi_gunhee choi_gunhee 4096 6월 7 19:32 u3_ctrl
drwxr-xr-x 2 choi_gunhee choi_gunhee 4096 3월 8 2022 Videos
```

```
choi_gunhee@gunhee-linux-94:~/test$ awk '{print $1, $9}' list.txt
total
drwxrwxr-x anaconda3
-rw-rw-r-- Anaconda3-2021.11-Linux-x86_64.sh
drwxrwxr-x cache-trace
-rw-rw-r-- cscope.files
-rw-rw-r-- cscope.out
drwxrwxr-x data
drwxr-xr-x Desktop
drwxr-xr-x Documents
drwxr-xr-x Downloads
drwxrwxr-x Easy-Shell-Script
drwxrwxr-x kangaroo_project
drwxrwxr-x memcached-1.6.16
-rw-rw-r-- memcached-1.6.16.tar.gz
drwxrwxr-x MyProject
-rw-rw-r-- nohup.out
drwxr-xr-x Pictures
drwxr-xr-x Public
drwxr-xr-x snap
drwxr-xr-x Templates
drwxrwxr-x test
drwxr-xr-x u3_ctrl
drwxr-xr-x Videos
choi_gunhee@gunhee-linux-94:~/test$
```

- User 계정 만들기

```
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd | grep choi_gunhee | wc -l
1
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd | grep choi_gunhee
choi_gunhee:x:1000:1000:choi_gunhee,,,:/home/choi_gunhee:/bin/bash
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

- User 계정 만들기
  -

```
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd
1
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd
choi_gunhee:x:1000:1000:choi_gunhee,,,:/home/choi_gunhee:/bin/bash
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

```
choi_gunhee@gunhee-linux-94:~/test$ cat addUser_script.sh
#!/bin/bash

if [[-n $1]] && [[-n $2]]
then
 UserId=$1
 Password=$2

 if [[$(cat /etc/passwd | grep $UserId | wc -l) == 0]]
 then
 useradd -m $UserId
 echo $Password | passwd $UserId --stdin
 else
 echo "this user is existing : $UserId"
 fi
else
 echo -e "Please input user id and password"
fi

choi_gunhee@gunhee-linux-94:~/test$ █
```

- 사용자별 사용량 체크

- ```
choi_gunhee@gunhee-linux-94:~/test$ cat ./userUsage.sh
```

```
#!/bin/bash
```

```
dir=/home
```

```
for userPath in $dir/*
```

```
do
```

```
    user=${userPath##*/}
```

```
    echo $user
```

```
    size=$(ls -l /home/$user | grep $user | awk '{ sum += $5 } END { print sum }')
```

```
    if [ -z $size ]
```

```
    then
```

```
        echo $user size is 0
```

```
    else
```

```
        echo $user size is $size
```

```
    fi
```

```
    echo ""
```

```
done
```

```
choi_gunhee@gunhee-linux-94:~/test$ █
```