

SERVER PROGRAMMING PRACTICE 2

CSLAB
Jo, Minwoo

Config files

일반적인 config 구조

- 블록 구조

- head

- [head]로 구성된 문장은 해당 블록을 의미

- body

- 변수 = 값으로 구성되어 있음
 - 프로그램은 선언되지 않은 변수를 인식
 - » 이럴 경우, 프로그램 내부 변수를 기본값으로 설정함

- All-in-One

- 한블럭 안에서 전체를 정의

```
[defaults]
base_features = sparse_super,large_file,filetype,resize_inode,dir_index,ext_attr
default_mntopts = acl,user_xattr
enable_periodic_fsck = 0
blocksize = 4096
inode_size = 256
inode_ratio = 16384

[fs_types]
ext3 = {
    features = has_journal
}
ext4 = {
    features = has_journal,extent,huge_file,flex_bg,metadata_csum,64bit,dir_nlink,extra_isize
}
rhel6_ext4 = {
    features = has_journal,extent,huge_file,flex_bg,uninit_bg,dir_nlink,extra_isize
    inode_size = 256
    enable_periodic_fsck = 1
    default_mntopts = ""
}
```

```
# To modify the system-wide sshd configuration, create a *.conf file under
# /etc/ssh/sshd_config.d/ which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 22
Port 122
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress :::
```

Arguments

프로그램 파라미터

- 파라미터
 - 타겟이나 옵션을 프로그램으로 입력하는데 사용
- Option
 - 선택적임 (필수가 아님)
 - 프로그램의 실행 구조를 변경

```
minwoo@cslab-rocky:~$ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
  or: cp [OPTION]... SOURCE... DIRECTORY
  or: cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
-a, --archive                same as -dR --preserve=all
  --attributes-only          don't copy the file data, just the attributes
  --backup[=CONTROL]        make a backup of each existing destination file
-b                            like --backup but does not accept an argument
  --copy-contents            copy contents of special files when recursive
-d                            same as --no-dereference --preserve=links
-f, --force                  if an existing destination file cannot be
```

Today's Goal

Part 1 : 계산기 만들기

- 단순 사칙연산 계산기 제작
 - WITH using arguments

Part 2 : 계산기 만들기 v2

- 2개의 모드 : sum / avg
- 옵션 1개 : 수 전체 표시(default)
 - WITH using config file

Part 3 : 로그인, 구구단 출력

- 로그인 기능 구현
- 로그인 성공 시 구구단 출력

Part 1. 계산기 만들기

1. 다음 요구사항에 맞춰 계산기를 제작하시오

요구사항

- 프로그램 파라미터를 통해 입력받아야 함
 - 입력은 연산자와 두 수 (e.g. ./calc add 1 3)
 - 출력은 예시와 같음 (e.g. 1 + 3 = 4)
 - 사칙연산 : add, sub, mul, div
 - 입력이 없을 경우 실행하지 않고, 매뉴얼을 출력해야함
 - Makefile을 이용한 컴파일을 해야함
2. redirection을 이용하여 텍스트로 출력할 것

Part 2. 계산기 만들기 v2

1. 다음 요구사항에 맞춰 계산기를 제작하시오

요구사항

- 프로그램 파라미터를 통해 입력받아야 함
 - 입력은 여러 수 (e.g. ./calc 1 2 3 4 5)
- 프로그램은 "calc.config"라는 config 파일을 참조해야함
- calc.config

```
[general]
mode = sum / avg
showAll = True(default) / False
```

- Makefile을 이용한 컴파일을 해야함

2. redirection을 이용하여 텍스트로 출력할 것

Part 3. 로그인, 구구단 출력

1. 다음 요구사항에 맞춰 프로그램을 제작하시오

요구사항

- 프로그램 파라미터를 통해 입력받아야 함
 - 입력은 문자열(ID) 혹은 None (e.g. ./prog admin // ./prog)
- 프로그램은 "passwd.config"라는 config 파일을 참조해야함
- passwd.config

```
[general]
sleep = 5
gugu = 3
[user]
id = server
passwd = programming
[admin]
passwd = 1q2w3e4r5t!
```

- 이 프로그램은 3번의 로그인 시도를 허용하며, 초과시 [sleep] 초만큼 sleep.
- 성공한다면, 구구단을 출력하는 프로세스를 [gugu]개 실행함
- Makefile을 이용한 컴파일을 해야함

Thank you