

Linux



IndiFlex 시니어코딩

Linux

1991, Linus Torvalds(Finn) from Unix

Linux Distro

- Debian : Debian, **Ubuntu**, KNOPPIX(CD Linux)
- Red Hat : Fedora, RedHat Enterprise, **CentOS**, Vine Linux(Jp)
- Slackware: openSUSE(Novell), SUSE Linux Enterprise

Kernel

- Management the Memory, File System, CPU, Device, etc.

Linux Shell

Manipulate Linux Kernel

Linux Shell Types

- **sh** (Bourne shell) : By Unix Shell, Super shell
- **bash** (Bourne-again shell) : Super shell in Linux
- **csh** (C shell) : C like syntax
- **tcsh** (Enhanced-C shell): c
- **ksh** (korn shell) : by David Korn, Powerful Script Language
- **zsh** (Z shell) : Unix/GNU shell script, Powerful Script Language

Linux File System Directories

Filesystem Hierarchy Standard

/bin : 기본 명령어

/boot : for booting

/dev : device file, cd-rom

/etc : config, passwd, rc.d

/home : user home dir

/lib : shared library

/media : ssd

/opt : application software package

/proc : process info

/root : root home dir

/sbin : 관리자용, ifconfig

/srv : system data

/tmp : temporary dir

/usr : source or programs

/usr/local

/var : logs, ftp, spool, mail

/lost+found

cf. **inode** (**ls -il**)

Linux Ports

IANA (Internet Assigned Numbers Authority)

20	FTP (data)
21	FTP (Control)
22	SSH / rsync / rcp
23	Telnet
25	SMTP (Simple Mail Transfer)
465	SMTPS
43	whois
53	DNS

80	HTTP
443	HTTPS
110	POP3
995	POP3S
123	NTP (Network Time Protocol)
143	IMAP2/4
993	IMAPS
514	SysLog

Command Line Tips

\$> tab → |

\$> Arrow Up & Down

\$> !

\$> !!

\$> Ctrl + A, Ctrl + E

\$> history

\$> man <명령>

한글 적용 (Root)

```
$> locale                # 현재 언어 설정
$> locale -a             # 적용 가능한 언어만 보기

$> apt-get update
$> apt-get install locales
$> cat /usr/share/i18n/SUPPORTED

$> localedef -f UTF-8 -i ko_KR ko_KR.UTF-8
$> locale -a

# ~/.profile (docker의 경우 .bashrc)에 추가
LC_ALL=ko_KR.UTF-8 bash
export LANGUAGE=ko
```

```
$> dnf install glibc-langpack-ko -y

if error!!
yum install glibc-locale-source glibc-langpack-en

# system locale setting
$> localedef -f UTF-8 -i ko_KR ko_KR.utf8

# (참고)
$> file -bi <file-name>    # yum install file
$> iconv -c -f euc-kr -t utf-8 a.js > a1.js
```

DNF

```
$> dnf install <pkg> -y
$> dnf update <pkg>
$> dnf remove <pkg>
$> dnf info <pkg>
```

기본 명령어

(참고) [putty 사용하기](#)

ls / touch / cat

head / tail / less

pwd / which / clear / echo

cd / cd -

mkdir / rmdir

cp / mv / rm

find

whoami

passwd

기본 명령어2

cf. nmon, dstat, nstat, sar, sysstat

df

du

free

top

vmstat

ps

ps -ef | grep bash

기본 명령어3

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

```
echo "aaaaaaaaaa"
```

chmod

chown

ln

ln -s 목적지(존재하는) 링크명

Try This

tt.sh를 ttt폴더로 옮기고,
파일명을 aa.sh로 변경하시오.

ubuntu

```
yum install java-1.8.0-openjdk
```

```
sudo apt-get update
```

```
sudo apt-get install openjdk-8-jdk -y
```

```
yum install java-1.8.0-openjdk -y
```

root command

/etc/passwd
/etc/shadow
/etc/group

```
$> su          # cf. su -
```

```
$> useradd -g <group> <userid>
```

```
#> groupadd <group>
```

```
$> passwd <account-name>
```

```
$> usermod <new-id> <old-id>
```

```
$> deluser <account-name>
```

```
cf. chage -d 0 <account-name>
```

grep

```
export GREP_OPTIONS='--color=always'
```

<https://regex101.com/>

```
grep <찾을단어> <file-name> [-io]
```

```
grep ^If zen.txt
```

```
grep idea.$ zen.txt
```

```
echo Two aa tao. | grep -i t[ow]o
```

```
echo "aaa 123 hi 459" | grep [0-9]
```

```
echo "Two twoo aa too." | grep -io two*
```

```
echo "Two __33__ two" | grep -io __.*__
```

```
echo "I love $ hh" | grep \$
```

```
cf. egrep <찾을단어> *.gz
```

Try This

주의) 1번 외, 개인계정과
queen계정만 사용할 것!!

1. queen 이라는 유저를 생성.
2. queen 계정의 홈 디렉토리에 test 디렉토리 생성
3. 개인계정 홈디렉토리에 queen의 test 폴더를 연결
4. 개인계정으로 연결된 test폴더에 df 명령의 결과를 담은 df.txt 파일 생성



Linux



vi editor

IndiFlex 시니어코딩

vi editor

command mode

|

edit mode

mode 전환 (esc <> i,a,A,o,O)

arrows (h, j, k, l, w, ctrl+f, ctrl+u/b, HML)

cw, dw, shift+A, shift+^, shift+\$, O

v, y, yy, p, dd, u, x, X, ., Shift+D

Ctrl+r, Ctrl+e, Ctrl+y

colon command (명령모드)

- w, q, x, !, /, %s, :n, :\$

- set nu, set nonu

- :set paste

search (명령모드)

- /검색어 → n, N

3

Linux



python, telnet, git 설치

IndiFlex 시니어코딩

Install Python & Pip in Ubuntu

```
$> sudo apt-get update
```

```
$> sudo apt-get install python3
```

```
$> sudo apt-get install python3-pip -y
```

```
$> which python3
```

```
$> mv /usr/bin/python3 /usr/bin/python
```

ntp

```
#> apt-get install ntp
```

```
ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime
```

Telnet Daemon in Ubuntu

`#!/bin/sh`

```
$> sudo apt-get install xinetd telnetd
```

```
$> vi /etc/xinetd.d/telnet
```

```
service telnet
```

```
{
```

```
    disable = no
```

```
    flags = REUSE
```

```
    socket_type = stream
```

```
    wait = no
```

```
    user = root
```

```
    server = /usr/sbin/in.telnetd
```

```
    log_on_failure += USERID
```

```
}
```

```
$> /etc/init.d/xinetd restart
```

Telnet Daemon in Ubuntu (Cont'd)

```
$> docker commit ub ub_telnet
```

```
$> docker run -itd -p 23:23 --name ubt ub_telnet bash
```

putty로 telnet 접속하기

git

```
#> apt-get install git
```

```
#> yum install git -y
```

```
#> git config --list
```

```
#> git config --global user.name <github-username>
```

```
#> git config --global user.email <email>
```

```
#> git config credential.helper store
```

```
#> git clone <github-url>
```



Linux



Docker Linux 구동 Tip

IndiFlex 시니어코딩

Tip: Container 실행시 자동으로 Telnet 띄우기

1. ubt container에서 **/bin/docker_bash** 셸 스크립트 작성

```
#!/bin/sh  
/etc/init.d/xinetd restart  
export LANGUAGE=ko  
LC_ALL=ko_KR.UTF-8 bash
```

2. **#> chmod +x /bin/docker_bash**

3. container 밖으로 나와서, Image 생성

```
$> docker commit ubt ubx
```

4. ubx 이미지를 이용하여 신규 컨테이너 생성 및 구동

```
$> docker run -itd --restart=always --name ubx -p 23:23 ubx /bin/docker_bash
```

5. 이제부터 ubx 사용!

```
$> docker attach ubx
```

6. ubx가 정상이면, 기존 ubt container 삭제

```
$> docker stop utb; docker rm utb
```

5

Linux



Shell Script / Cron

IndiFlex 시니어코딩

~/.vimrc

bash

```
set tabstop=4
set shiftwidth=4
set expandtab
set smartindent
set autoindent
set number
set showmatch
```

shell script

bash

```
#!/bin/sh 또는  #!/bin/bash
```

```
echo "Hello Shell Script!"  
printf "%s %s %d\n" aa bb 123
```

```
str="hello "  
echo "${str}world"  
echo "$str world"
```

loop

bash

```
#!/bin/sh 또는  #!/bin/bash
```

```
# 1 ~ 100 loop
```

```
for i in {1..100}; do ~ done
```

```
# infinite loop
```

```
while true; do ~; done
```

Try This

현재 폴더의 *.txt 파일들의
내용을 한번에 출력하는
스크립트를 작성하시오.

```
#!/bin/bash

for i in `ls *.txt`
do
    echo " ----- $i"
    cat $i
    echo "===== "
done
```

Variables

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

```
echo "$0 $1 $#"
```

```
STR="abc"
```

```
echo $STR
```

```
i=100
```

```
ii=$(( $i + 100 ))
```

```
echo "i=${i}, ii=${ii}"
```

IF

문자: == !=

숫자: -gt, eq, lt, ne, le, ge

파일: -f, -d, -r, -w, -x

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

```
if [ $# -gt 0 ]; then  
    cat $1
```

```
else
```

```
    echo "Input filename.."
```

```
fi
```

```
if [ -f t.txt ]; then  
    cat t.txt
```

```
fi
```


Try This

파일명을 입력받으면
해당 파일의 내용을 출력하는
스크립트를 작성하시오.
(단, 파일명 미입력시 안내 메시지
출력)

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

```
if [ $# -eq 0 ]; then  
    echo "Input the filename, please.."  
    echo "usage) ./s4.sh <file>"  
    exit 0  
fi
```

```
cat $1
```

date

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

```
DATE=`date +%Y-%m-%d`
```

```
echo $DATE
```

```
DATE=`date +%Y-%m-%d --date=yesterday`
```

```
DATE=`date +%Y-%m-%d --date='1 day ago'`
```

```
DATE=`date +%Y-%m-%d --date='2 day ago'`
```

```
DATE=`date +%Y-%m-%d --date='2 day'`
```

```
DT=`date +%Y-%m-%d --date='1 week ago'`
```

```
DT=`date +%Y-%m-%d --date='1 month ago'`
```

```
DT=`date +%Y-%m-%d --date='1 month'`
```

Try This

파일명을 입력받으면
오늘날짜(연월일).txt로
파일명 변경

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

```
if [ $# -eq 0 ]; then  
    echo "Input the filename, please.."  
    echo "usage) ./s5.sh  
<to-change-file>"  
    exit 0  
fi
```

```
DATE=`date +%Y%m%d`  
FN="${DATE}.txt"  
#echo "mv $1 $FN"
```

```
mv $1 $FN
```

Try This

파일명을 2개 입력 받아,
두 파일을 합친 후,
파일명을 **어제날짜.log** 라고
새롭게 만드시오.

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

```
if [ $# -lt 2 ]; then  
    echo "Input 2 files, please.."  
    exit 0  
fi
```

```
DATE=`date +%Y%m%d --date=yesterday`  
FN="${DATE}.log"
```

```
cat $1 > $FN
```

```
cat $2 >> $FN
```

Try This

구구단 출력

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

```
for i in {2..9}
do
```

```
    for j in {1..9}
    do
```

```
        echo "$i * $j = $(( $i * $j ))"
```

```
    done
```

```
    echo
```

```
"-----"
```

```
done
```

Array

배열, list

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

```
declare -a arr
```

```
arr=("aaa" "bbb" "ccc" 123)
```

```
echo $arr
```

```
echo ${arr[0]}
```

```
arr[4]="6666666"
```

```
echo ${arr[@]}
```

```
echo "${#arr} : ${#arr[@]}"
```

```
for i in ${arr[@]}; do ...
```

Function

```
$> ./s9.sh aaa
```

```
./s9.sh aaa 1
```

```
Hello ./s9.sh Jade Jeon by Jeon!! (2)
```

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

```
echo "$0 $@ $#"
```

```
say_hello() {  
    echo "Hello $0 $@ by $2!!  
    ($#)"  
}
```

```
say_hello "Jade" "Jeon"
```

IFS & AWK

Internal Field Separator

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

```
echo "IFS=${IFS}."
```

```
PRE_IFS=$IFS
```

```
IFS="
```

```
"
```

```
for i in `ls -al`; do
```

```
    echo $i | awk '{print $5}'
```

```
done
```

```
IFS=$PRE_IFS
```


Try This

/bin 디렉토리 아래에 존재하는
파일들의 이름과 크기를
한개의 파일로 생성하시오.
(단, total , .. 등은 제외)

```
#!/bin/bash

PRE_IFS=$IFS

IFS="
"

cd /home/dooo

FileName="bin_files.txt"
touch $FileName

echo " ----- "
TOT=0
for i in `ls -al /bin`; do
    S=`echo $i | awk '{print $5}'`
    F=`echo $i | awk '{print $9}'`

    if [ "$F" == "." ] || [ "$F" == ".." ] || [ "$F" == "" ]; then
        continue
    # elif
    fi

    #TOT=$(( $TOT + $S ))
    TOT=`expr $TOT + $S`

    echo "$S $F" >> $FileName
done

echo "Total Size is $TOT"

IFS=$PRE_IFS
```

cron

- **CentOS**
yum install crontie
systemctl restart crond
(crond [start | stop])
- ubuntu
apt-get update
apt-get install cron
service cron restart

```
$> crontab -l
```

```
$> crontab -e
```

분 시 일 월 주

```
* * * * * /test.sh >> /temp.log 2>&1
```

(2: Standard Error, 1: Standard Out)

```
$> ps -ef | grep cron
```