

일일 업무 사항 정리

| | |
|-------|------------|
| 작성자 | 제품팀 이민성 인턴 |
| 업무 일시 | 2022.08.22 |

세부 사항

1. 업무 내역 요약 정리

| 목표 내역 | Done & Plan | To-do |
|--|---|---|
| 1. 리눅스 환경에 대해 이해 2. 리눅스 명령어 숙달 3. 계정 및 디렉토리 관리 공부 4. 셸 스크립트 언어 이해 및 활용 5. 오라클 데이터베이스 설치과정 복기 | <p>오늘은 리눅스를 여러 방면으로 활용하려고 시도했습니다. 활용 예제가 뭐가 있을까 곰곰이 생각해보다가 제가 실제로 윈도우 환경에서 사용하는 프로그램들을 설치하면 좋겠다고 생각했습니다. 먼저, 인터넷 실행을 해보기 위해 기본적으로 설치되어 있는 firefox를 실행해서 인터넷에 접속해보았습니다. 둘째, 제가 평상시에 주로 사용하는 브라우저인 크롬을 설치하고 실행해보았습니다. 셋째, 제가 주로 사용하는 메신저인 카카오톡을 설치하고 실행해보려 했지만 오류가 발생하였습니다. 수정 방법을 찾아도 잘 모르겠어서 리눅스에 대해 좀 더 공부해보고 나중에 해보겠습니다.</p> <p>그리고 \$에 대해 교육을 받았습니다. 쓸 일이 없다고 생각하였었지만 쓸 일이 많다는 것을 깨달았습니다. 그리고 디렉터리 경로 (상대경로, 절대경로), x윈도시스템, xftp, xshell, xming, ftp, sftp, 프로토콜 등에 대해 알게 되었습니다.</p> <p>셸에 대해 알아보았으며 내일은 셸 스크립트 언어에 대해 더 공부하고 예제를 통해 활용하고 버추얼 박스를 이용해 리눅스를 설치할 예정입니다. 그리고 차후에 오라클 데이터베이스 설치과정을 간단하게 복기해 볼 예정입니다.</p> | <ul style="list-style-type: none"> - 리눅스 환경에 대한 이해 -- 계정 및 디렉토리 관리 -- 기본 명령어 - 셸 스크립트 언어에 대한 이해 -- 셸과 셸 스크립트 언어 정의 -- 기본 문법 - 설치 -- 과정 복기 |

2. 내용 세부 (업무 세부 내역 정리 및 기타 사항 정리)

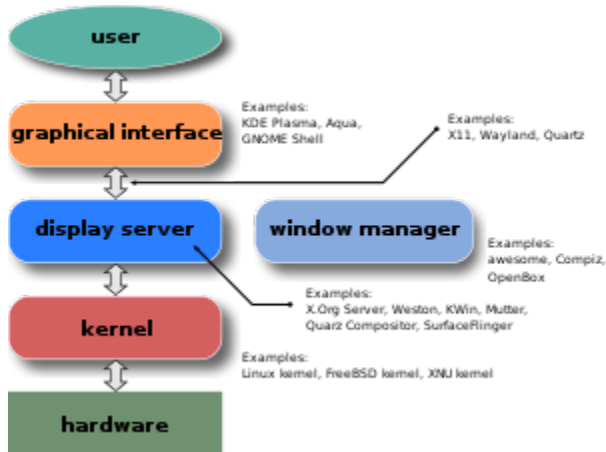
1. xshell

Xshell : 넷사랑컴퓨터에서 개발한 터미널 에뮬레이터로 SSH, 텔넷 등의 프로토콜을 지원한다.

2. xming

Xming : 윈도우 XP이상의 마이크로소프트 윈도우 운영 체제를 위한 X11* 디스플레이 서버이다.

*X11 (x 윈도우시스템) : 유닉스 계열 운영체제에서 사용되는 윈도 시스템 및 X 윈도우 GUI 환경



3. xftp

Xftp : 넷사랑컴퓨터에서 개발한 FTP*, SFTP* 클라이언트 프로그램이다

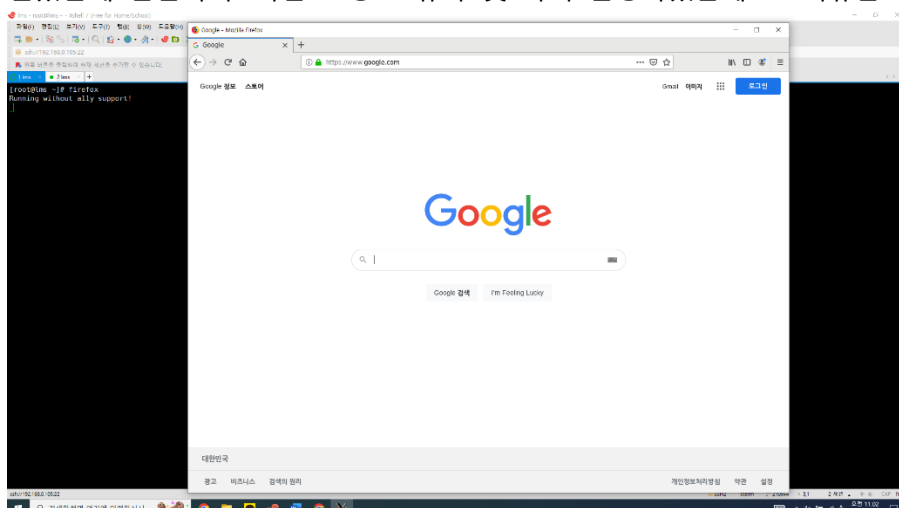
ftp : 파일 전송 프로토콜(File Transfer Protocol)의 약자로 인터넷상의 장치가 파일을 전송할 때 사용하는 규칙이다.

프로토콜 : 전자기기가 서로 통신하는 데 필요한 절차나 규칙

*sftp : SSH 파일 전송 프로토콜로 시큐어 셸 데이터 스트림을 통해 보안 등급이 높은 파일 전송을 실현하는 별도의 프로토콜이다.

Linux CentOS7 cli환경에서 google 접속하기

대부분의 리눅스에는 마이크로소프트 윈도우의 옛지처럼 firefox가 기본적으로 설치되어있다. 그래서 google을 접속하기 위해서는 터미널에 firefox라는 명령어를 치면 xming을 통해서 firefox에 접속된다. 굉장히 복잡할 줄 알았는데 간단하다. 하는 도중 오류가 몇 가지 발생하였는데 그 이유는 root 계정으로 접속하지 않았기 때문이다.



Linux CentOS7 cli환경에서 크롬 설치하고 접속하기

1. wget*명령을 사용하여 최신 Google Chrome .rpm 패키지를 다운로드 한다.

*wget : 웹 서버에서 파일을 다운로드 명령어

```
[root@lms ~]# wget https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm
```

2. 파일이 다운로드되면 Google Chrome을 CentOS7 시스템에 입력한다.

```
[root@lms ~]# yum localinstall google-chrome-stable_current_x86_64.rpm
```

이후 chrome 명령어를 치면 알 수 없는 명령어라고 뜨는 것을 볼 수 있다.

```
[root@lms ~]# chrome
bash: chrome: 명령을 찾을 수 없습니다 ...
```

이는 실행파일의 위치를 정의해 놓은 환경변수에 크롬의 위치가 정의되지 않았기 때문이다. 따라서 크롬을 명령어로 만들어야 한다. 크롬파일을 이미 환경변수에 등록되어 있는 폴더에 옮기면 된다.

3. 크롬 실행 파일의 위치는 /opt/google/chrome이다. 폴더 이동 후 ls를 쳐보면 아래와 같이 녹색으로 되어있는 파일이 보이는데 이것이 실행파일이다.

```
[root@lms ~]# cd /opt/google/chrome
[root@lms chrome]# ls
MEIPreload                libnspr4.so.0d          product_logo_128.png
WidevineCdm               libnss3.so.1d           product_logo_16.png
chrome                   libnssutil3.so.1d       product_logo_24.png
chrome-management-service liboptimization_guide_internal.so product_logo_256.png
chrome-sandbox            libplc4.so.0d           product_logo_32.png
chrome_100_percent.pak    libplds4.so.0d          product_logo_32.xpm
chrome_200_percent.pak    libsmime3.so.1d         product_logo_48.png
chrome_crashpad_handler   libssl3.so.1d           product_logo_64.png
default-app-block         libvk_swiftshader.so    resources.pak
default_apps              libvulkan.so.1          v8_context_snapshot.bin
google-chrome             locales                 vk_swiftshader_icd.json
icudtl.dat               nacl_helper             xdg-mime
libEGL.so                nacl_helper_bootstrap   xdg-settings
libGLESv2.so              nacl_irt_x86_64.nexe
```

이를 실행시키기 위해서는 절대경로/상대경로를 써줘야한다.

절대경로는 최상위 폴더인 루트부터 경로로 자신의 위치가 어디든 상관없이 쓰일 수 있다.

```
[root@lms ~]# /opt/google/chrome/chrome
```

상대경로는 현재의 폴더를 기점으로 한 경로이다. 만약 현재 실행파일이 들어있는 폴더의 위치에 있다면

```
[root@lms chrome]# ./chrome
```

여기서 ./중 .는 현재 자신의 폴더를 의미한다. 따라서 ./는 "자신의 폴더 아래에 있는"으로 해석할 수 있다.

- 실행 중 아래와 같은 오류가 발생하였다면

```
[root@lms chrome]# ./chrome
[10016:10016:0822/125043.898651:ERROR:zygote_host_impl_linux.cc(90)] Running as root without --no-sandbox is not supported. See https://crbug.com/638180.
```

아래의 경로로 들어가서 --no-sandbox를 맨 아래에 추가해주면 오류가 발생하지 않는다.

```
[root@lms chrome]# vi /opt/google/chrome/google-chrome
```

```
exec -a "$0" "$HERE/chrome" "$@" --no-sandbox
```

위와 같은 방법으로 하면 실행되기는 하지만 실행할 때마다 저렇게 긴 경로를 적으면 불편하다.

4. /opt/google/chrome/에 있는 chrome 파일의 심볼릭링크 파일을 어플리케이션이 담겨있는 /usr/bin에 만들어 놓으면 된다. ln에 앞에는 절대경로를 적어주고, 뒤에는 파일을 만들고자 하는 위치를 적어주면 된다.

*ln 명령어 : 링크 파일을 만들 때 사용하는 명령어

*ln -s : 심볼릭 링크 파일을 생성한다

```
[root@lms chrome]# ln -s /opt/google/chrome/chrome /usr/bin/chrome
```

5. [root@lms chrome]# ls -altr /usr/bin | grep chrome

```
lrwxrwxrwx. 1 root root 25 8월 22 13:02 chrome -> /opt/google/chrome/chrome
```

/usr/bin/에 chrome이라는 파일이 생성된 것을 볼 수 있다.

그럼 이제 어느 위치에 있는 chrome를 사용하면 크롬이 열리는 것을 확인할 수 있다.

또한 chrome 위에 웹페이지 주소를 적으면 바로 그 페이지가 열린다.

하지만 위와 같은 오류가 발생하여서 고쳤는데 또 다시 같은 오류가 발생한다면 google-chrome명령어로 실행해주면 된다.

Linux CentOS7 에 카카오톡 설치하기

1. Linux에서는 기본적으로 윈도우에서 쓰는 *.exe파일을 사용할 수 없다. 이 단점을 보완하기 위한 것이 wine*이다.

*wine : 윈도우가 아닌 타 운영체제에서도 윈도우 기반 소프트웨어를 실행할 수 있게 해주는 프로그램
yum을 이용하여 wine을 설치한다.

```
[root@lms /]# yum install wine
```

2. 카카오톡 다운로드 홈페이지에 들어간 후 *.exe파일을 다운받고 ctrl+alt+f를 눌러 xftp를 실행 시킨 후 root/다운로드 경로에 옮겨준다.

| 이름 | 상태 | 진행률 | 크기 | 로컬 경로 | <--> 원격 경로 | 속도 | 남은 시간 | 경과 시간 |
|-------------------------|------|-----|-----------------|-------------------------|---------------------------|-----------|----------|----------|
| [K] KakaoTalk_Setup.exe | 진행 중 | 93% | 73.56MB/78.47MB | C:\Users\Wchoco\Down... | 192.168.0.105:/root/다운... | 9.76 MB/s | 00:00:01 | 00:00:06 |

192.168.0.105:22에 연결됨. 이진 1 개체가 선택됨 78.47MB

3. root/다운로드 경로로 들어간다.

```
[root@lms /]# cd root
[root@lms ~]# ls
anaconda-ks.cfg google-chrome-stable_current_x86_64.rpm initial-setup-ks.cfg 공 개 다 운 로 드 문 서 박 탕 화 면 비 디 오 사 진 서 식 음 악
[root@lms ~]# cd 다 운 로 드
[root@lms 다운로드]# ls
```

4. wine을 이용하여 카카오톡을 다운로드 받는다. 최초 wine 실행 시 wine mono installer과 wine gecko installer을 설치하게 된다. 설치를 묻는 창이 나오면 모두 다 설치를 누른다.

5. 설치가 완료된다.

- 나의 경우에는 아래와 같은 오류가 발생하였다. 여러 방면으로 원인을 찾아봤지만 고치는 방법을 잘 모르겠다. 일단 shell과 다른 것들을 공부한 후 나중에 다시 해보아야겠다.

```
[root@lms 다운로드]# wine KakaoTalk_Setup.exe
wine: 잘못된 Z:\root\다운로드\KakaoTalk_Setup.exe 실행 형식 .
```

8.19 질문 리뷰

Q : @testuser의 /sc/road.txt를 /notepad로 옮기려고 [testuser@lms sc]\$ mv road.txt /notepad를 했는데 mv: cannot move `road.txt' to `/notepad': 허가 거부 라고 에러가 발생합니다. 원인이 무엇일까요?

A : [testuser@lms sc]\$ mv road.txt /notepad 가 실행되지 않은 이유는 경로 문제이다. notepad 디렉터리 경로는 sc에 있지 않고 /home/testuser/notepad에 있다. 위와 같은 명령으로 실행하게 되면 디렉토리를 /home/testuser/sc/notepad으로 인식하게된다. 따라서 디렉토리 경로를 /home/testuser/notepad로 하면 된다.

```
[testuser@lms ~]$ tree
.
├── notepad
│   ├── silcroad.txt
│   ├── test.txt
│   └── tree.txt
└── sc
    ├── road.txt
    └── silc.txt
```

```
[testuser@lms sc]$ mv road.txt /home/testuser/notepad
```

리눅스 셸

- \$: 셸 변수 (아래와 같은 예시처럼 변수가 무엇인지 기억이 나지 않을 때 많이 사용한다.)

```
i=100
echo "i:$i"
```

(test.sh 파일)

- 셸 : 운영 체제 상에서 다양한 운영 체제 기능과 서비스를 구현하는 인터페이스를 제공하는 프로그램이다. 사용자와 커널 사이의 인터페이스를 감싸는 층으로 일반적으로 명령 줄(CLI)와 그래픽 형 (GUI) 2 종류로 분류된다.
- 셸 스크립트 : 셸로 하는 프로그래밍이다. 자바나 C언어처럼 조건문, 반복문 등의 일반적인 프로그래밍 문법을 사용할 수 있다. 그러나 컴파일 방식이 아닌 인터프리터를 통한 스크립트 방식이라는 차이점이 있다.

| 컴파일 언어 | 스크립트 언어 |
|--|---|
| 컴파일러를 거쳐서 기계어로 번역한 후 한 번에 실행 (번역 후 실행) | 인터프리터가 코드를 한줄한줄 해석하며 실행 (번역하며 실행) |
| 컴파일을 하기 위해 변수 선언 등의 제약사항이 많음 | 변수를 변수라고 선언하지도 않으며 해당 변수 자료형은 소스코드가 실행되는 순간임 |
| 코드 문법 등의 버그가 있으면 컴파일 자체가 안 됨 | 코드 실행 전까지 버그를 인지할 수 없기 때문에 오류 발견시점이 늦어짐 (디버깅이 까다롭다) |
| Java, C, C++ 등 | Java Script, Python 등 |

- 셸 스크립트를 사용하는 경우 : 서버가 다운되어서 확인해보니 서버 저장공간이 하나도 남아 있지 않았다. 이유는 로그 파일이 많이 쌓였기 때문이었다. 로그 파일이 업데이트가 안되어 관련 프로그램이 비정상적으로 종료되었다. 어떻게 하면 자동으로 오래된, 혹은 일정 시간 경과한 로그 파일을 삭제할 수 있을까? 이런 문제를 간단한 셸 스크립트 생성 및 주기적 실행으로 해결할 수 있다.

- 셸 스크립트 기본 문법

1. 셸 스크립트는 파일로 작성 후, 파일을 실행한다.
2. 파일의 가장 위 첫 라인은 `#!/bin/bash`로 시작한다.
3. 셸 스크립트 파일은 코드를 작성한 후에는 실행 권한을 부여해야한다.
4. 일반적으로 파일이름.sh 와 같은 형태로 파일 이름을 작성한다.
5. 주석은 `#` 으로 처리한다.

- ex1. bash 셸에서 제공하는 echo 함수를 이용하여 화면에 "Hello bash!"를 출력할 수 있도록 스크립트를 작성해보자 (echo는 화면에 출력해주는 셸 명령이다.)

```
[testuser@lms sh]$ vi hello.sh
[testuser@lms sh]$ ls -l
합 계 4
-rw-rw-r--. 1 testuser testuser 31  8월 22 18:05 hello.sh
[testuser@lms sh]$ chmod 777 hello.sh
```

```
#!/bin/bash

echo "Hello Bash!"
```

vi로 파일을 생성한 후 chmod로 권한을 부여한다.

```
[testuser@lms sh]$ sh hello.sh
Hello Bash!
```

sh명령어로 실행할 수 있다.

- 변수

- ◆ 선언 : 변수명=데이터 (변수명=데이터 사이에 띄어쓰기는 허용되지 않는다.)
- ◆ 사용 : \$변수명 (echo \$변수명\$변수명)
- ◆ 코드 예시

```
[testuser@lms sh]$ vi variable.sh

#!/bin/bash

mysql_id='root'
mysql_directory='/etc/mysql'

echo $mysql_id
echo $mysql_directory

[testuser@lms sh]$ sh variable.sh
root
/etc/mysql
```

- ex2. 아이디 관련 정보 변수를 만들어보자 (이름, 나이, 직업)+

```
[testuser@lms sh]$ vi id.sh  
[testuser@lms sh]$ sh id.sh  
minsung20student
```

```
#!/bin/bash  
  
name='minsung'  
age=20  
carrer='student'  
  
echo $name$age$carrer
```