

1차 과제 - 리눅스 기초

학 과 : 컴퓨터 공학과
담당교수 : 황호영
분 반 : 목34
학 번 : 2016722092
성 명 : 정동호

목차

1. Ubuntu Installation
 2. Usage of Linux Commands
-

Ubuntu Installation

Introduction

리눅스 기초를 다지기 위해 가장 대중적이고 진입장벽이 낮은 Ubuntu 설치부터 시작한다. 그리고 설치 과정을 캡처하고 설명한다.

Result

기존에 이미 Ubuntu 16.04.03 가상 호스트가 있었지만 과제 수행을 위해 다시 설치하였다. 하드웨어 설정은 다음과 같다:

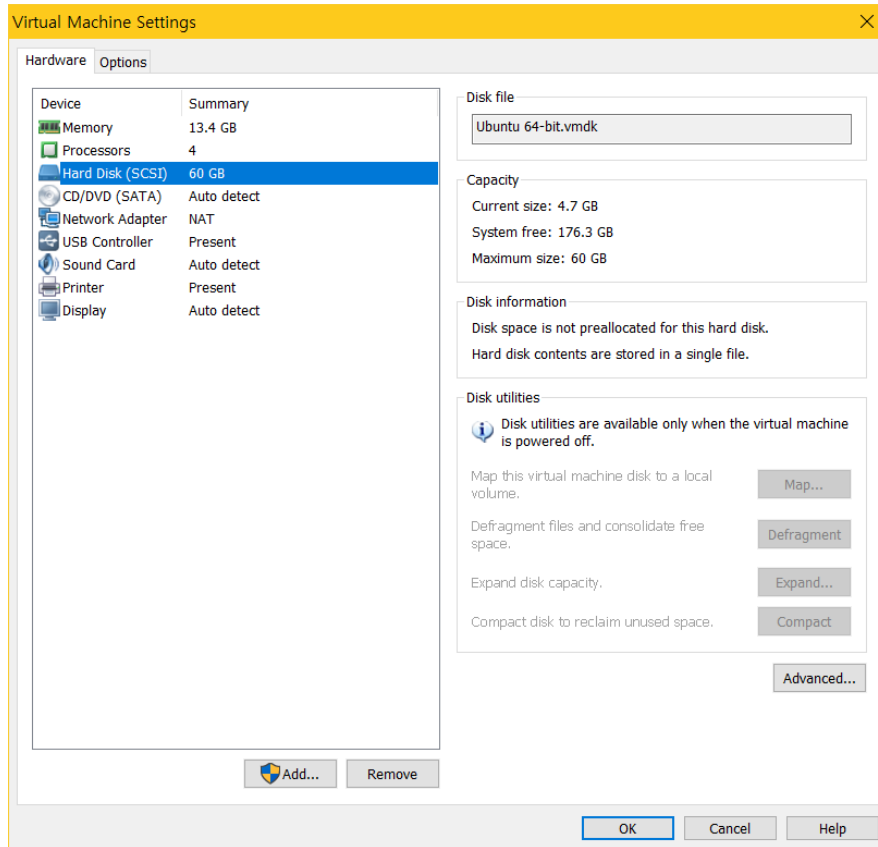


그림 1-1. 하드웨어 설정

다음은 설치 과정 중 언어 팩을 다운로드 하는 장면이다:

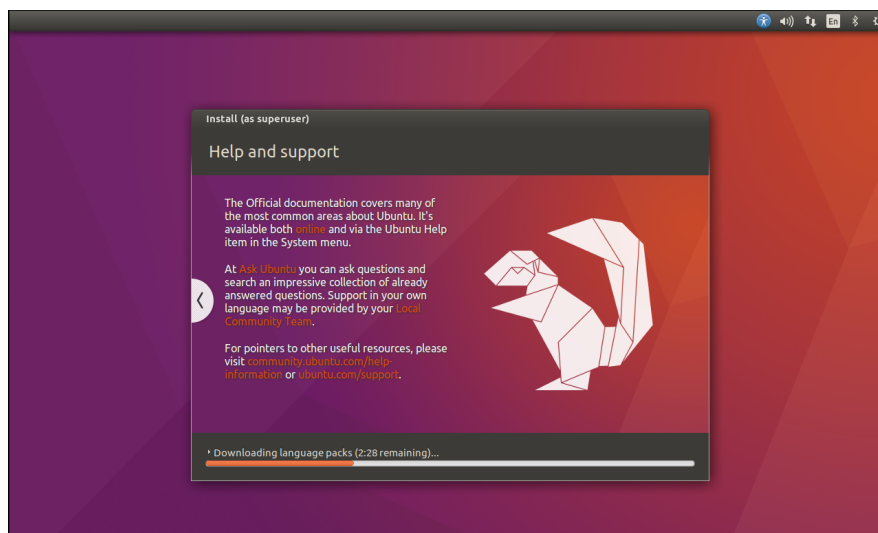


그림 1-2. 언어 팩 다운로드

다음과 같이 깔끔한 초기 화면을 볼 수 있었다:



그림 1-3. 초기 화면

Reference

<https://www.ubuntu.com/download/desktop> — Ubuntu 16.04.4 LTS (Xenial Xerus) 이미지 제공

Usage of Linux Commands

Introduction

우분투를 설치하고 난 후 리눅스의 기본적인 명령어들은 어떤 것들이 있는 지 알아보고 직접 터미널에서 따라 쳐 보면서 다양한 명령어와 옵션들을 실습한다. 실습에서 사용할 명령어들은 다음과 같다:

- 2주차 : `man`, `cat`, `pwd`, `cd`, `ls`, `chmod`, `mkdir`, `rmdir`, `rm`, `cp`, `mv`, `ln`, `touch`, `exit`, `kill`, `ps`, `pstree`, `time`, `passwd`, `uname`, `wc`, `more`, `echo`, `alias`, `find`, `grep`
- 3주차 : `vi`, `make`, `gdb`

2주차

1. `man`

```
dongho@ubuntu: ~  
WALL(1) User Commands WALL(1)  
  
NAME  
    wall - write a message to all users  
  
SYNOPSIS  
    wall [-n] [-t timeout] [message | file]  
  
DESCRIPTION  
    wall displays a message, or the contents of a file, or otherwise its  
    standard input, on the terminals of all currently logged in users. The  
    command will wrap lines that are longer than 79 characters. Short  
    lines are whitespace padded to have 79 characters. The command will  
    always put a carriage return and new line at the end of each line.  
  
    Only the superuser can write on the terminals of users who have chosen  
    to deny messages or are using a program which automatically denies mes-  
    sages.  
  
    Reading from a file is refused when the invoker is not superuser and  
    the program is suid or sgid.  
  
OPTIONS  
Manual page wall(1) line 1 (press h for help or q to quit)
```

그림 2-1-1. man wall

```
dongho@ubuntu: ~  
dongho@ubuntu:~$ man -k copy  
Clone (3pm) - recursively copy Perl datatypes  
bcopy (3) - copy byte sequence  
copysign (3) - copy sign of a number  
copysignf (3) - copy sign of a number  
copysignl (3) - copy sign of a number  
cp (1) - copy files and directories  
cpgr (8) - copy with locking the given file to the password or gr...  
cpio (1) - copy files to and from archives  
cppw (8) - copy with locking the given file to the password or gr...  
dd (1) - convert and copy a file  
debconf-copydb (1) - copy a debconf database  
File::Copy::Recursive (3pm) - Perl extension for recursively copying files an...  
getunwind (2) - copy the unwind data to caller's buffer  
getutmp (3) - copy utmp structure to utmpx, and vice versa  
getutmpx (3) - copy utmp structure to utmpx, and vice versa  
gvfs-copy (1) - Copy files  
gvfs-move (1) - Copy files  
install (1) - copy files and set attributes  
mcopy (1) - copy MSDOS files to/from Unix  
memccpy (3) - copy memory area  
memcpy (3) - copy memory area  
memmove (3) - copy memory area  
mempcpy (3) - copy memory area  
ntfscp (8) - copy file to an NTFS volume.  
objcopy (1) - copy and translate object files  
rcp (1) - secure copy (remote file copy program)  
rsync (1) - a fast, versatile, remote (and local) file-copying tool  
scp (1) - secure copy (remote file copy program)  
ssh-copy-id (1) - use locally available keys to authorise logins on a re...  
stpcpy (3) - copy a string returning a pointer to its end  
stpncpy (3) - copy a fixed-size string, returning a pointer to its end  
strcpy (3) - copy a string  
strncpy (3) - copy a string  
va_copy (3) - variable argument lists  
wcpncpy (3) - copy a wide-character string, returning a pointer to i...  
wcpncpy (3) - copy a fixed-size string of wide characters, returning...  
wcscpy (3) - copy a wide-character string  
wcsncpy (3) - copy a fixed-size string of wide characters  
wmemcpy (3) - copy an array of wide-characters  
wmemmove (3) - copy an array of wide-characters  
wmemcpy (3) - copy memory area  
x86_64-linux-gnu-objcopy (1) - copy and translate object files  
dongho@ubuntu:~$
```

그림 2-1-2. man -k copy

```
dongho@ubuntu: ~  
WRITE(1) BSD General Commands Manual WRITE(1)  
  
NAME  
    write - send a message to another user  
  
SYNOPSIS  
    write user [tty]  
  
DESCRIPTION  
    The write utility allows you to communicate with other users, by copying  
    lines from your terminal to theirs.  
  
    When you run the write command, the user you are writing to gets a mes-  
    sage of the form:  
  
        Message from yourname@yourhost on yourtty at hh:mm ...  
  
    Any further lines you enter will be copied to the specified user's termi-  
    nal. If the other user wants to reply, they must run write as well.  
  
    When you are done, type an end-of-file or interrupt character. The other  
    user will see the message 'EOF' indicating that the conversation is over.  
  
    You can prevent people (other than the super-user) from writing to you  
    with the mesg(1) command.  
  
Manual page write(1) line 1 (press h for help or q to quit)  
dongho@ubuntu: ~  
dongho@ubuntu:~$ man -a write  
--Man-- next: write(2) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]  
dongho@ubuntu: ~  
WRITE(2) Linux Programmer's Manual WRITE(2)  
  
NAME  
    write - write to a file descriptor  
  
SYNOPSIS  
    #include <unistd.h>  
  
    ssize_t write(int fd, const void *buf, size_t count);  
  
DESCRIPTION  
    write() writes up to count bytes from the buffer pointed buf to the  
    file referred to by the file descriptor fd.  
  
Manual page write(2) line 1 (press h for help or q to quit)
```

그림 2-1-3. man -a write

먼저 `man [OPTION...][SECTION] PAGE...` 에서 옵션 없이 `wall` 을 인자로 주었을 때는 `wall` 에 대한 설명이 나타나는 것을 볼 수 있었다. 그리고 `-k` 옵션을 주고서 `copy` 를 인자로 주었을때는 `copy` 가 포함된 모든 페이지들의 리스트가 반환되었다. 마지막으로 man에는 여러 섹션이 있고, 동일한 이름의 페이지가 존재할 수 있는데 `[SECTION]`을 명시하지 않고 `a` 옵션을 준다면 그림 2-1-3과 같이 한 페이지의 조회가 끝나면 다른 페이지도 조회할 것인지 선택할 수 있다는 것을 볼 수 있었다.

man의 섹션은 일반적으로 다음과 같이 8가지로 나뉜다:

섹션	설명
1	일반적인 명령어
2	시스템 콜
3	라이브러리 함수—특히 C 표준 라이브러리의
4	특수 파일(보통 /dev에서 볼 수 있는 장치)과 드라이버
5	파일포맷 및 규약
6	게임 및 화면보호기
7	잡다한 것들
8	시스템 관리 명령어 및 대몬

모든 man page들은 일관성과 간결함을 위해 공통된 양식을 따르고 있다. 이 양식을 통해 페이지들은 가능한한 하이라이팅이나 폰트제어 없이 ASCII만으로 출력하는데 최적화 되고 있다. 양식은 다음과 같다:

- NAME : 명령어 또는 함수의 이름, 뒤이어 무엇을 하는지 한줄 짜리 설명이 나온다.
- SYNOPSIS : 명령어의 경우 어떻게 실행시키는지에 대한 설명을 출력하고, 프로그램 함수의 경우 함수가 취하는 인자들과 해당 함수를 정의한 헤더 파일들에 대한 목록을 나열한다.
- DESCRIPTION : 명령어 또는 함수의 기능에 대한 설명
- EXAMPLES : 일반적인 용례
- SEE ALSO : 관련된 명령어나 함수

물론 위의 절뿐만 아니라 OPTIONS, EXIT STATUS, ENVIRONMENT, ... 등도 포함할 수 있다.