한양대학교 HANYANG UNIVERSITY

Linux Programming Overview

System Programming

Index

I. 리눅스의 기본 개념

리눅스의 기본 개념

❑ 다중 작업 (Multi Process)

- 선점 가능한 (Preemptive) 실제 다중 작업을 지원
- 작업은 실행 중인 프로그램을 의미 (프로세스)

❑ 다중 사용자 (Multi User)

- 동시에 여러 명의 사용자가 시스템에서 작업하는 것을 허용
- 터미널이나 네트워크 연결을 통해서 동일한 하나의 리눅스 시스템을 사용

❑ 다중 프로세서 (multi processor)

- 다중 프로세서 구조에서도 실행될 수 있음 (두 개 이상의 CPU를 가진 시스템을 리눅스가 지원)

❑ 이식성과 확장성

- 이식성이 높음
- 다양한 언어로 작성된 프로그램을 어렵지 않게 사용할 수 있음

I. 리눅스의 기본 개념

리눅스의 기본 개념

❑ 파일 시스템

- 리눅스의 파일 시스템은 유닉스의 것과 같이 트리 구조를 이루고 있음
- 별도로 추가된 물리적인 보조 기억 장치들이나 하드웨어 디바이스들도 파일 형태로 파일 시스템에 연결

❑ 권한

- 사용자 별로 별도의 권한을 부여
- 하나의 시스템을 여러 명의 사용자가 동시에 사용할 수 있기 때문에 발생할 수 있는 여러 가지 문제를 사전에 방지
- 시스템을 관리하기 위한 관리자와 시스템을 사용하기만 하는 사용자

❑ 셸

- 사용자가 시스템을 쉽게 사용할 수 있도록 중간자 역할의 프로그램
- 사용자가 명령어 라인을 입력해서 원하는 작업을 수행할 수 있음

❑ 개발 환경

- 리눅스는 새로운 프로그램을 개발하기 위한 환경을 제공
- 프로그래밍 언어용 컴파일러, 프로그램 개발에 필요한 보조적인 유틸리티

리눅스 시스템

❑ 리눅스의 명령어 사용하기

- 리눅스의 파일 시스템은 유닉스의 것과 같이 트리 구조를 이루고 있음
- 별도로 추가된 물리적인 보조 기억 장치들이나 하드웨어 디바이스들도 파일 형태로 파일 시스템에 연결

```
lee@ubuntu: ~
lee@ubuntu:~$ id
uid=1000(lee) gid=1000(lee) groups=1000(lee),4(adm),24(cdrom),27(sudo),30(dip),4
6(plugdev),108(lpadmin),124(sambashare)
lee@ubuntu:~$
```

사용자의 ID와 그룹을 확인하는 "id" 명령어의 사용 예

```
lee@ubuntu: ~
ID(1)                          User Commands                          ID(1)

NAME
       id - print real and effective user and group IDs

SYNOPSIS
       id [OPTION]... [USERNAME]

DESCRIPTION
       Print  user  and group information for the specified USERNAME, or (when
       USERNAME omitted) for the current user.

       -a     ignore, for compatibility with other versions

       -Z, --context
              print only the security context of the current user

       -g, --group
              print only the effective group ID

       -G, --groups
              print all group IDs

Manual page id(1) line 1 (press h for help or q to quit)
```

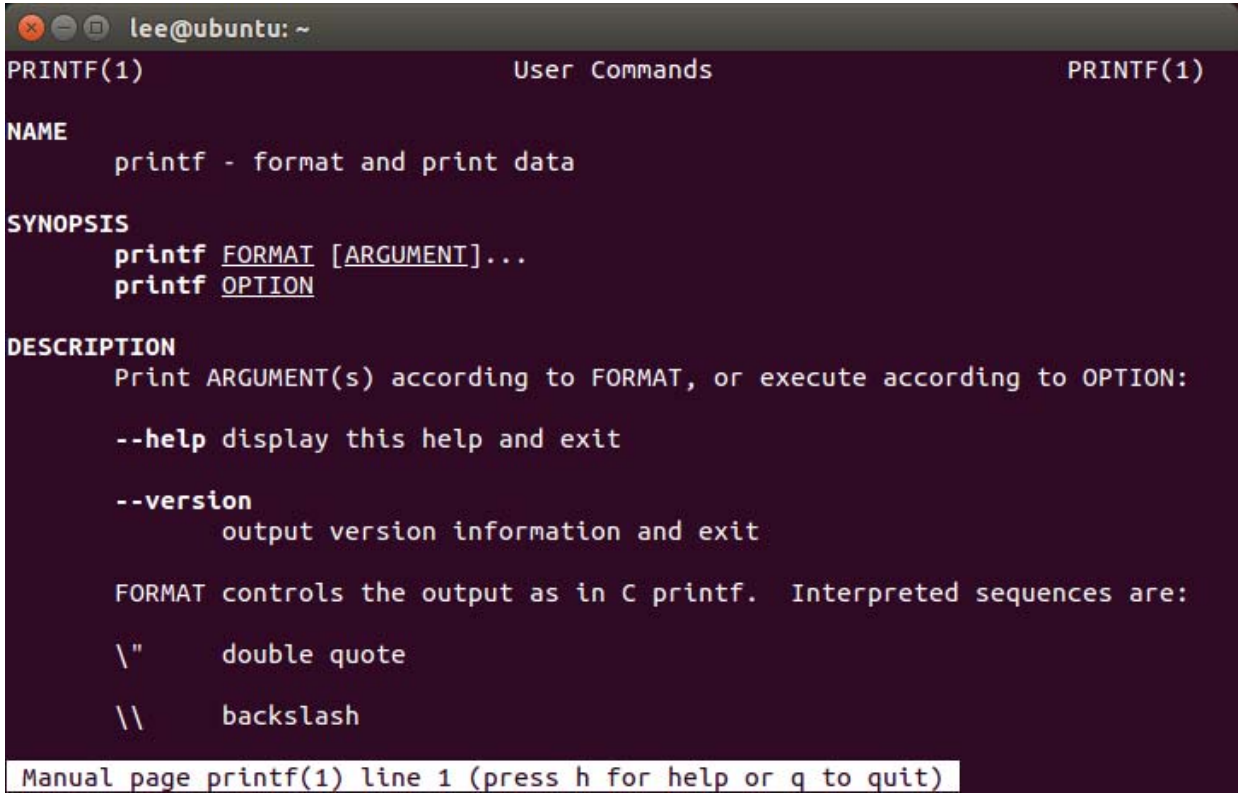
"man id"를 실행하여 id 명령의 사용법을 확인하는 예

한양대학교 HANYANG UNIVERSITY

리눅스 시스템

❑ man 명령어의 사용 예

- man 명령어는 리눅스 명령어, 프로그래밍 언어의 함수, 셸 스크립트 등의 사용법도 확인 가능함
- "man printf"를 실행하여 printf의 사용법 확인



man 명령은 시스템의 명령이나 프로그래밍 언어의 함수 사용법을 익히기 위한 핵심이다.

한양대학교 HANYANG UNIVERSITY

III. 리눅스의 프로그래밍 환경

프로그래밍 관련 툴

❑ 에디터 (editor)

- 소스 코드를 편집하는 용도로 사용됨
- 일반적으로 vi 에디터를 사용함
- gedit : GUI 에디터, 사용이 쉬움

❑ 컴파일러 (compiler)

- 소스 코드를 바이너리 코드로 변경
- 리눅스 시스템에서는 gcc를 사용

❑ 링커/로더 (linker/loader)

- 목적(object) 파일들을 연결해서 실행 파일을 생성
- 리눅스 시스템에서는 ld를 사용
- 대부분 컴파일러가 알아서 실행

소스 코드 편집하기 (1/2)

❑ 소스 코드를 편집하기 위한 두 가지 방법

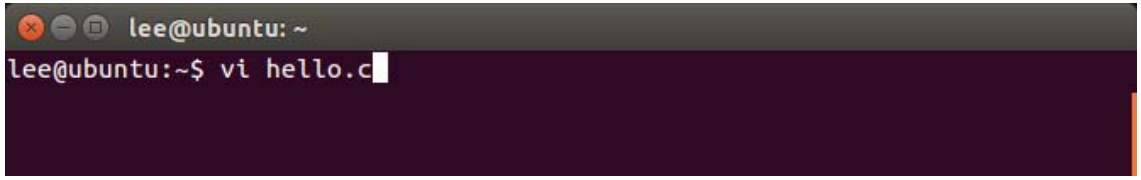
- PC에서 편집하여 FTP 서비스로 리눅스 시스템에 업로드
- 리눅스 시스템에서 직접 편집

❑ 리눅스 시스템에서 직접 편집

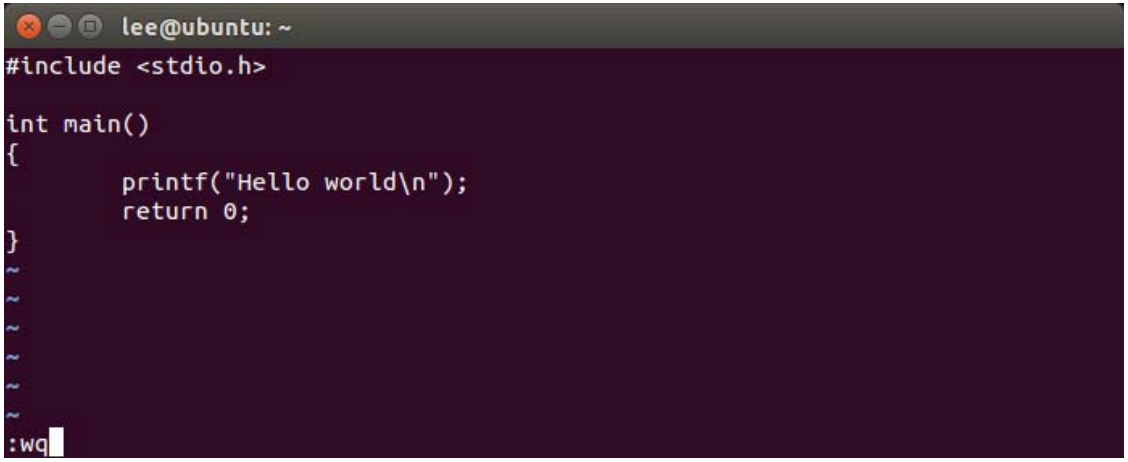
- 간단한 소스 코드
  - $ cat > hello.c
- 복잡한 소스 코드
  - $ vi hello.c
- vi 편집기를 사용하여 소스 코드를 편집하는 것이 일반적

소스 코드 편집하기 (2/2)

❑ vi를 사용한 소스 코드 편집

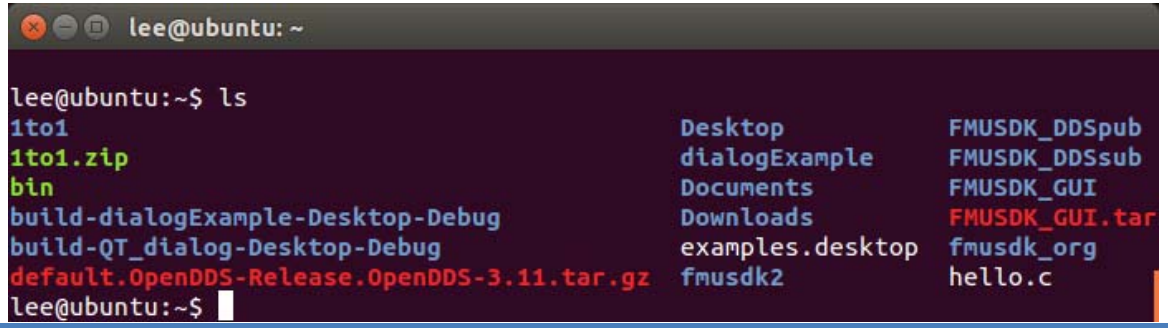


← vi 에디터를 실행



← 소스 코드를 편집

← 소스 코드를 저장



← 저장된 소스 코드 확인

gcc 사용하기

❑ $ gcc [options] source_files

- options

  -o output_filename : 실행파일을 만들 경우 실행 파일의 이름을 지정한다.

  -c : 지정한 소스코드의 목적(object) 파일을 만든다.

- source_files

  .c 를 확장자로 가지는 소스 코드(들)

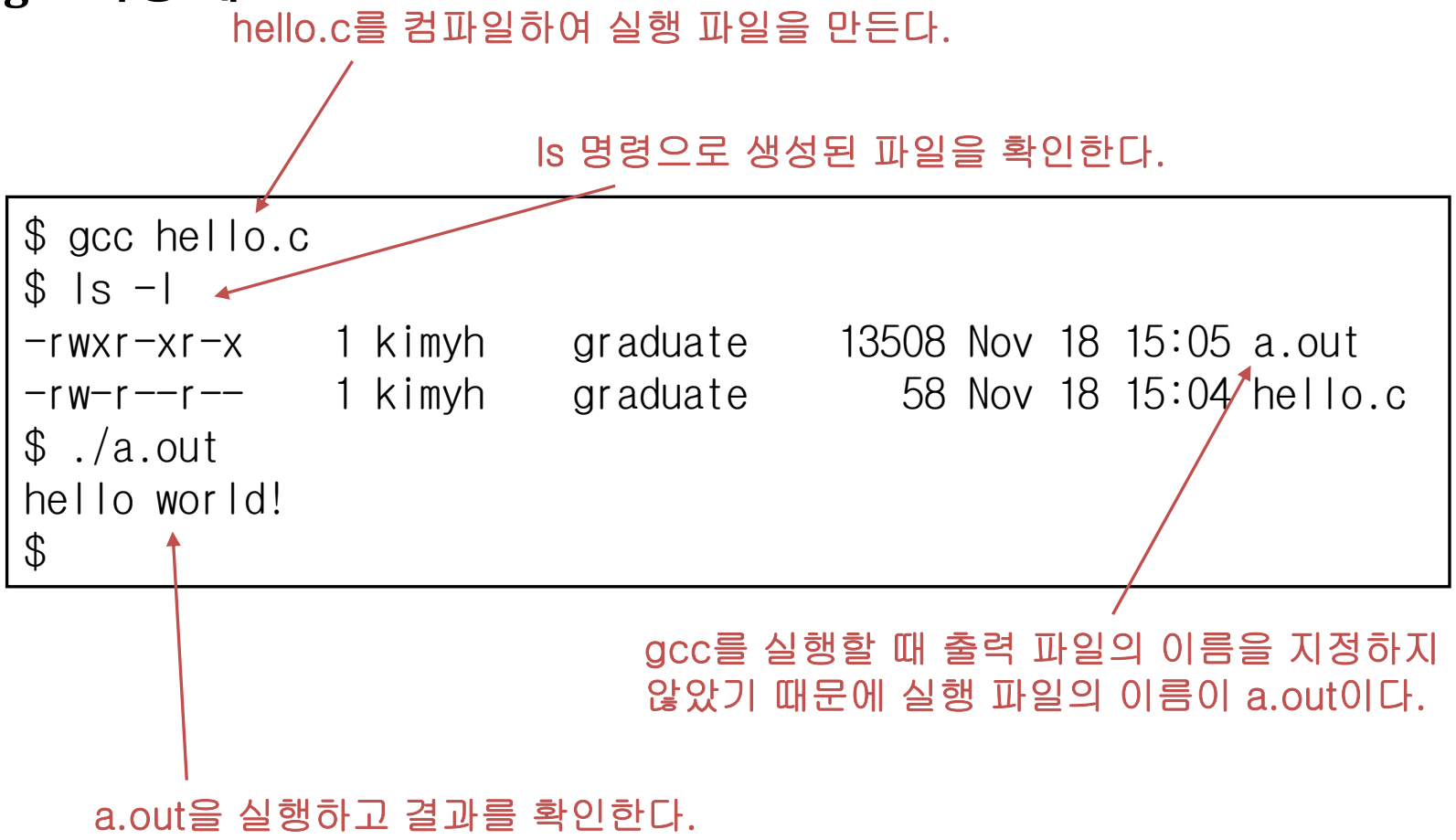
- 사용 예

  $ gcc hello.c

  $ gcc –o hello hello.c

gcc 사용하기

❑ gcc 사용 예



gcc 사용하기

❑ 컴파일 시 오류 발생 및 해결

– 소스 코드를 컴파일할 때 오류가 있을 경우 gcc는 오류 메시지를 출력한다.

```
$ cat hello.c
#include <stdio.h>



main()
{
        printf("hello world!₩n")
}
$ gcc -o hello hello.c
hello.c: In function `main':
hello.c:6: parse error before `}'
$
```

문장이 ';'으로 종결되지 않았다.

컴파일을 한 결과 소스 코드의 6번 라인에서 오류가 생겼다.

gcc 사용하기

❑ 두 개 이상의 소스 코드로 하나의 실행 파일 만들기

– 큰 규모의 프로그램은 여러 개의 소스 코드로 나누어서 작성하는 것이 일반적이다.

one.c	two.c

#include <stdio.h>
void printmsg(void);
main()
{
printmsg();
}
	#include <stdio.h>
	void printmsg(void)
	{
	printf("hello world!₩n");
	}

```
$ gcc –o three one.c two.c
```

출력 파일의 이름은 three이다.

소스 코드 파일은 2개로 각각 one.c와 two.c다.

gcc 사용하기

❑ 교재 소스코드 헤더파일 다운받기

- 아래 사이트에 접속해서 다운받기
- http://www.apuebook.com/code3e.html
- cp Downloads/src.3e.tar.gz ~/ (홈폴더로 이동)
- Tar –zxvf src.3e.tar.gz로 압축풀기
- 이후 자신이 원하는 디렉토리명으로 수정하기(원하면 그대로 사용 가능)
  - mv apue.3e APUE3(강사의 경우, APUE3로 변경)

❑ 전체 코드 컴파일

– sudo apt-get install libbsd-dev (bsd-dev라이브러리 설치)

– cd APUE3(폴더명) 후 make

gcc 사용하기

❑ 라이브러리와 헤더를 포함해서 컴파일하기

-I(대문자 아이) 옵션 : #include 문장에서 지정한 헤더 파일이 들어있는 곳을 정하는 옵션

-l(소문자 엘) 옵션 : 링크(link)할 라이브러리를 명시해주는 옵션

- Ex) libmylib.a 를 사용하는 경우 -lmylib으로 링크

만약, c 소스 코드에서 apue.h(**/home/sjhong/APUE3/include)**를 포함하고 libapue.a (**/home/sjhong/APUE3/lib**)라이브러리를 사용하는 경우에 c 소스코드를 컴파일하려면 아래와 같이 입력하면 된다.

```
$ gcc  –o  test test.c –I/home/sjhong/APUE3/include –lapue
–L/home/sjhong/APUE3/lib
```

gcc 사용하기

❑ 책자 예제 코드 컴파일 및 실행 해보기

- Prog. 1.3
- Prog. 1.4
- Prog. 1.5
- Prog. 1.6
- Prog. 1.7
- Prog. 1.10

Q and A