

라즈베리 파이를 활용한 IoT 프로젝트

라즈베리 파이를 위한 리눅스 기초 배우기

2일차

담당교수: 조도은

<https://github.com/DoEunCho/raspberrypi>



1일차 : 라즈베리파이 소개와 환경 구축(3H)

2일차 : 라즈베리 파이를 위한 리눅스 기초 배우기(3H)

3일차 : 파이썬 기초 명령어 익히기(3H)

4일차 : 라즈베리 파이 GPIO와 센서 동작하기(3H)

5일차 : 나만의 가상비서 만들기(구글 어시스턴트)(3H)



■ 라즈베리 파이를 위한 리눅스 기초 배우기

- 리눅스 디렉터리 구조
- 리눅스 명령어 사용하기
 - 파일관리
 - 계정관리
 - 시스템 관리
 - 메모리 관리
 - 네트워크 관리
- 리눅스 에디터 사용하기



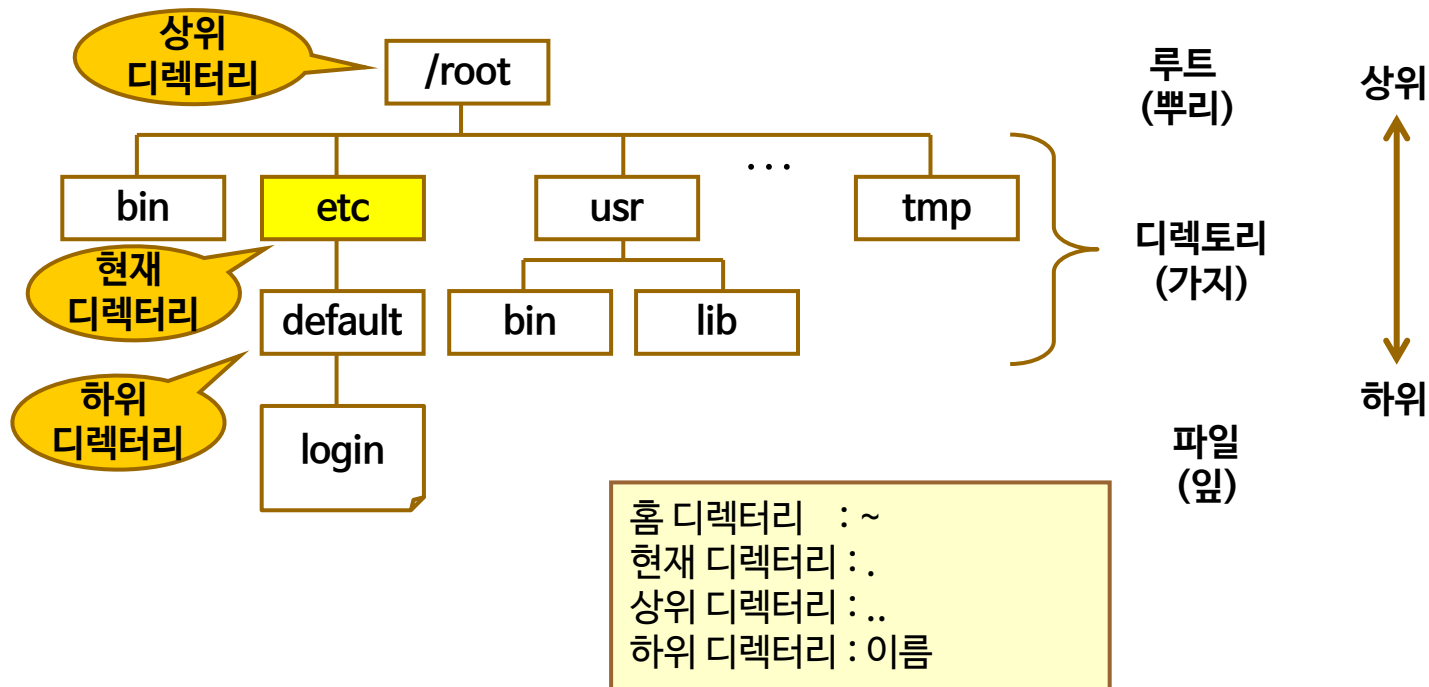
- 윈도우에서 **폴더**를 리눅스에서는 **디렉터리**라 부름
- 리눅스는 디렉터를 구분할 때 “/”를 사용
 - **/** : 최상위 디렉터리 (root)
 - **/bin** : 리눅스의 기본 명령어인 복사(cp), 삭제(rm), 이동(mv)같은 명령어가 저장
 - **/dev** : 각종 디바이스 파일들 disk, mem, serial 등 시스템 디바이스 파일이 저장
 - **/etc** : 시스템의 각종 설정 파일들이 존재하는 디렉터리
 - **/home** : 일반 사용자의 홈 디렉터리
 - **/lib** : 각종 라이브러리들이 존재하는 디렉터리



- **/root** : 시스템 관리자 홈 디렉터리
- **/usr** : 시스템에 사용되는 각종 프로그램들이 설치되는 디렉토리
- **/usr/bin** : 응용 프로그램 설치 시 일반 사용자 명령어가 추가됨
- **/sbin** : 시스템 운영에 필요한 명령어 위치
- **/usr/sbin** : 응용 프로그램 설치 시 관리자 명령어가 추가됨
- **/tmp** : 임시 파일을 저장하는 위치
- **/opt** : 크기가 큰 리눅스 패키지 설치함
- **/var/log** : 시스템 로그 파일
- **/var/spool/mail** : 서버의 메일 임시보관 장소
- **/lost+found** : 부팅할 때 파일 시스템에 이상이 있는지 진단, 파일 시스템 복구 후 복구된 파일이 저장됨

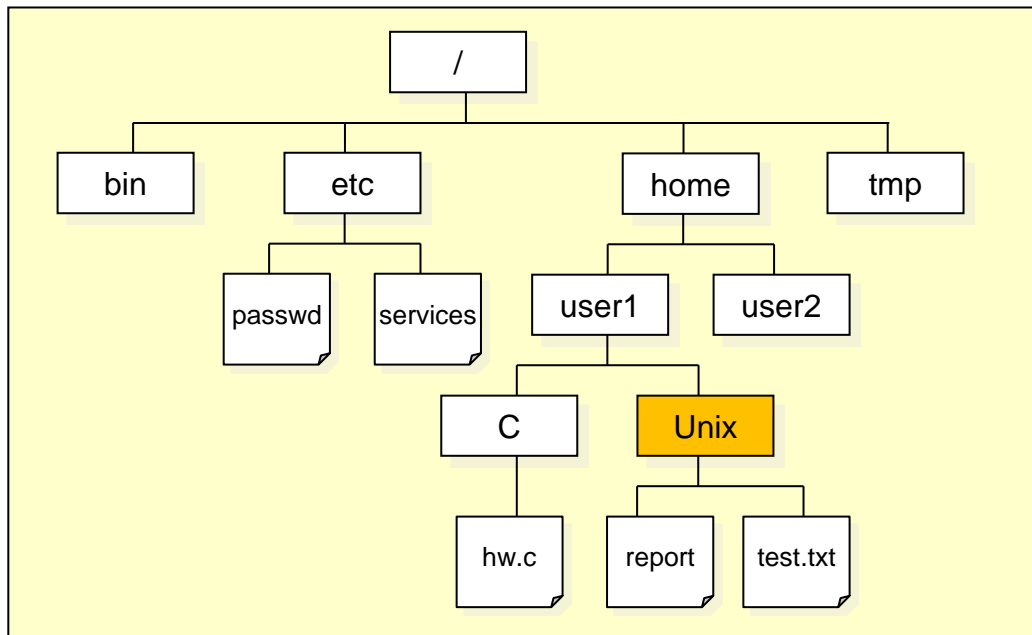


디렉터리 계층 구조



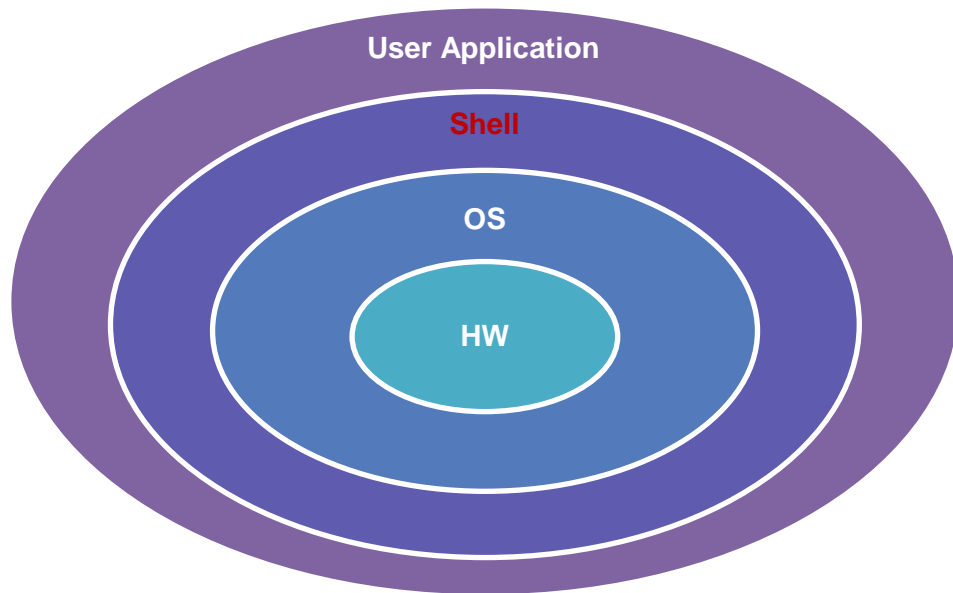


디렉터리 계층 구조



1. Unix의 절대경로 :
/home/user1/Unix

2. Unix의 상대경로 :
Unix



명령어 실행하기

- ~\$ **date** : 날짜 확인
- ~\$ **hostname** : 호스트 이름 확인
- ~\$ **ls** : 디렉터리 안의 목록 확인
- ~\$ **clear** : 화면 지우기



파일 관리 명령어

■ 파일 복사

- **cp** [옵션] [파일1] [파일2]
 - 옵션 -i: 파일2가 존재할 때 겹쳐 쓸지를 확인
 - 사용 예) \$ cp -i file1 file2

■ 파일 제거

- **rm** [옵션] [파일]
 - 옵션 -i: 각 파일에 대하여 지울 것인지 아닌지를 확인
 - r: 디렉터리와 그 안에 있는 모든 파일 전체를 재귀적으로 삭제
 - 사용 예) \$ rm -i file1

파일 복사하고 제거하기

```
~$ touch a.txt #파일 만들기
~$ ls -l
~$ cp -i a.txt b.txt
~$ ls -l
~$ rm -i b.txt
~$ ls -l
```

※ 간단한 단축키: 명령 종료 <ctrl+c>, 자동완성 <tab>, 이전명령 실행<up>



파일 관리 명령어

■ 파일 이동

- **mv** [파일1] [파일2 또는 디렉터리명]
 - 옵션 -b : 백업파일을 만든 후 파일을 지우거나 옮김
 - i : 이미 존재하는 파일을 덮어 쓸 것인지 물어봄
 - v : 파일을 옮기면서 파일명을 출력
 - 사용 예) \$ mv -i file1 file2
 - 사용 예) \$ mv -i file1 dir1

■ 디렉터리 만들기

- **mkdir** [디렉터리명]

■ 현재 위치확인

- **pwd**

파일 이동과 목록 확인하기

```
~$ mv -i a.txt hello.txt  
~$ ls -l
```

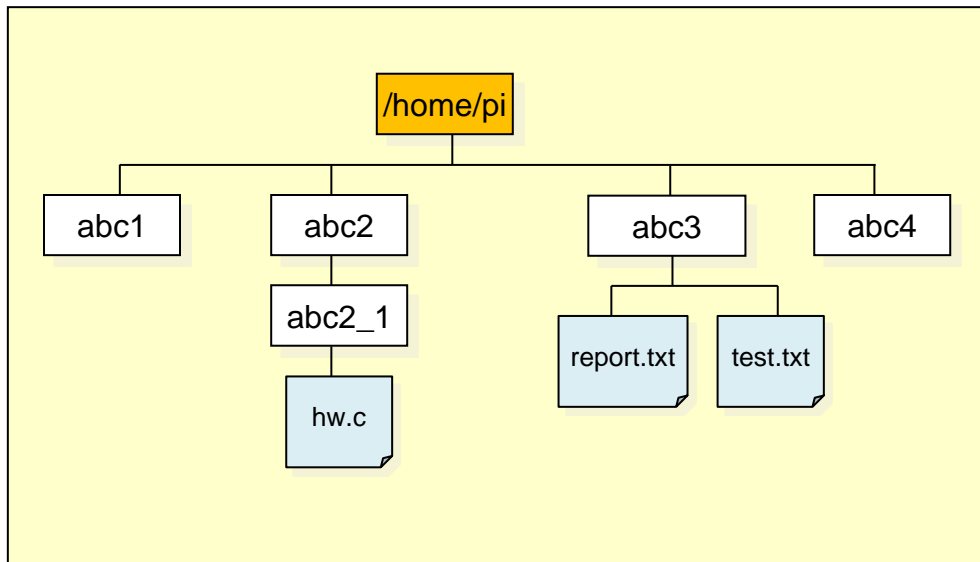
디렉터리 만들고 확인하기

```
~$ mkdir tmp  
~$ cd tmp  
~$ mkdir temp  
~$ cd temp  
~$ pwd
```



실습하기

- 주어진 그림과 같은 구조의 디렉터리와 파일을 추가 생성해 보세요





파일 관리 명령어

- 디렉터리 삭제
 - **rmkdir** [디렉터리명]
- 파일 리스트 화면 출력
 - **ls** [옵션] [파일 또는 디렉터리]
 - 옵션
 - l: 파일에 대한 정보(파일 허용권, 소유자, 그룹, 크기, 날짜)를 표시
 - a: 숨김 파일을 포함한 모든 파일을 화면에 표시
 - t: 최종 수정된 시간순으로 화면에 표시
 - u: 최근에 액세스했던 파일들을 보여줌
 - F: 파일의 특성을 보여줌
 - 디렉터리 : 이름/
 - 실행 파일 : 이름*
 - 심볼릭 링크 : 이름@
 - R: 하위 디렉터리의 파일도 모두 보여줌
 - 사용 예) \$ ls -tl

```
~$ cd ..  
~$ rmdir tmp  
~$ cd tmp  
~$ ls  
~$ rmdir temp  
~$ ls
```



■ ls -l 명령

```
$ ls -l
-rw-r--r-- 1 user1 staff 183 May 9 23:11 first.dat
```

① ② ③ ④ ⑤ ⑥ ⑦ ⑧

번호	값	의 미
①	-	파일 종류 (- : 일반파일, d: 디렉토리)
②	rw-r--r--	파일을 읽고,쓰고,실행할 수 있는 권한 표시
③	1	물리적 연결 개수
④	user1	파일 소유자의 사용자 명
⑤	staff	파일 소유자의 그룹명
⑥	183	파일 크기 (바이트 단위)
⑦	May 9 23:11	파일이 마지막으로 변경된 시간
⑧	first.dat	파일명



파일 관리 명령어

■ 파일 찾기

▣ **find** [파일 또는 디렉터리]

- 사용 예) **\$ find filename** : 파일을 현재 디렉터리에서 찾아 그 경로를 화면에 출력
- 사용 예) **\$ find / -name filename** : 최상위 루트 디렉터리부터 검색하여 검색된 파일을 한 행에 하나씩 화면에 출력
- 사용 예) **\$ find . -name filename** : 현재 디렉터리와 그 하위 디렉터리에서 찾을

■ 파일에서 특정 문장패턴(문자열) 찾기

▣ **grep** [문장패턴] [파일]

- 사용 예) **\$ grep on led.py** : on이 들어있는 라인을 화면에 출력
- 사용 예) **\$ ls -l | grep IoT** : 현재 디렉터리에 있는 파일 중 IoT를 찾아 출력



■ 현재 위치 변경

□ **cd**

- 사용 예) \$ cd IoT : 현재 위치를 IoT 디렉터리로 이동
- 사용 예) \$ cd .. : 상위 디렉터리로 이동
- 사용 예) \$ cd /IoT : 절대경로를 표시하여 IoT 디렉터리로 이동

■ 파일내용 화면 출력

□ **cat** [옵션] [파일1]

- 사용 예) \$ cat > input : control-D를 입력할 때까지 키보드에 입력한 내용을 input 파일에 저장
- 사용 예) \$ cat test1 test2 > test 3 : test1과 test2의 내용을 test3에 저장

※ con**CAT**enate : 파일 내용 화면 출력, 여러 개의 파일 하나로 합치기, 키보드 입력 파일 생성 등



```
~$ cat > text2    # 입력하는 내용을 text2 파일로 생성하기
```

```
Hi Linux!!
```

```
<Ctrl - D>
```

```
~$ cat text2
```

```
~$ clear
```

```
~$ cd /
```

```
~$ cd ~
```

```
~$ cd ..
```

```
~$ ls
```

```
~$ cat > text1
```

```
Welcome to Linux World!!
```

```
<Ctrl - D>
```




■ 파일 접근 권한

- 리눅스는 멀티 유저 시스템으로 여러 사용자가 시스템 접근이 가능
- 사용자는 하나 또는 여러 그룹에 속해 있으며, 파일에 사용자와 그룹의 접근 권한을 부여하여 접근을 제한
- 파일의 접근 권한은 `ls -l` 명령으로 확인

권한표시	파일 수	소유자	그룹	크기	변경 일자	파일명
drwxr-xr-x	2	pi	home	4096	May 27 08:31	bin

- d : 유형 (d:디렉터리, -:파일)
- r : read, w : write, x : execute, - : 권한 없음
- 2 : bin 디렉터리 내의 파일 수
- pi : 소유자
- home : 그룹
- 4096 : 크기
- May 27 08:31 : 수정한 날짜 및 시간
- bin : 파일 혹은 디렉터리 이름



- 디렉터리 권한 내용

- d rwx r-x r-x

- ① ② ③

- ① `rw`x : 파일 소유자 권한(파일소유자가 읽기, 쓰기, 실행 권한 있음)

- ② `r-x` : 그룹 권한(그룹이 읽기, 실행 권한 있음)

- ③ `r-x` : 기타 권한(기타 사용자가 읽기, 실행 권한 있음)

- 디렉터리 권한 변경

- **`chmod 755 filename`**

- 파일소유자는 모든 권한, 그룹과 기타 사용자는 읽기와 실행 권한만 부여

```
~$ ls -l
```

```
~$ chmod 755 test2
```

```
~$ ls -l
```



실습하기

1. test1파일의 접근 속성을 모든 사용자 읽고, 쓰고, 실행 할 수 있도록 변경합니다.
2. test2 파일의 접근 속성을 소유자는 읽고 쓰기만 가능하고 그룹 사용자와 다른 사용자는 읽기만 가능하도록 변경합니다.



계정관리

- 로그인
 - **login**
- 아이디
 - **id**
 - 사용 예) \$ id : uid=0(root) gid=0(root) groups=0(root)
- 비밀번호 변경
 - **passwd**
 - 사용 예) \$ passwd
Changing password for pi.
Current password:
New password:
Retype new password:
passwd: password updated successfully



■ 로그인한 사용자 정보 알기

□ who

- 옵션 : -u : 현재 로그인 된 사용자만을 나열

-H : 각 열 위에 헤더를 표시

- 사용 예) \$ who

```
pi    tty7    2020-05-27 14:42 (:0)
```

```
pi    tty1    2020-05-27 14:42
```

```
pi    pts/1    2020-05-28 07:38 (192.168.0.13)
```



기타 시스템 관리

■ 프로세스 리스트

□ ps

- 사용 예) \$ ps

PID	TTY	TIME	CMD
2296	pts/1	00:00:00	bash
3665	pts/1	00:00:00	ps

■ 프로세스 종료

□ exit

■ 프로세스 강제 종료

□ kill [프로세스 ID]

```
~$ ps
~$ vim test3
~$ ctrl-z
~$ vim ps
~$ ps
~$ kill -9 905(프로세스id)
~$ ps
~$ pkill -9 vim(프로세스 이름)
```



■ 별명 만들기

□ **alias**

- 사용 예) \$ alias mv = 'mv -i' : mv를 mv -i로 별명을 하나 더 만들
mv만 입력해도 겹쳐 쓸지 여부를 확인

```
~$ alias ls='ls -l'
```

```
~$ ls
```

■ 현재 시각 출력

□ **date**

■ 패키지 관리

□ **apt-get install** [패키지 이름]

□ **apt-get remove** [패키지 이름]

□ **apt-get - purge remove** [패키지 이름] : 패키지를 설정 파일을 포함하여 지움

```
~$ date
```



메모리 관리

- 메모리 용량 확인
 - **df**
- 메모리 상태 표시
 - **free**

```
pi@raspberrypi: ~  
pi@raspberrypi:~$ df  
Filesystem      1K-blocks    Used Available Use% Mounted on  
/dev/root        14989948 2966592  11363200  21% /  
devtmpfs         469532      0    469532    0% /dev  
tmpfs            474140      0    474140    0% /dev/shm  
tmpfs            474140    12304    461836    3% /run  
tmpfs             5120        4      5116    1% /run/lock  
tmpfs            474140      0    474140    0% /sys/fs/cgroup  
/dev/mmcblk0p1   258095     52145   205950   21% /boot  
tmpfs            94828      0    94828    0% /run/user/1000
```

```
pi@raspberrypi: ~  
pi@raspberrypi:~$ free  
              total        used        free      shared  buff/cache   available  
Mem:           948280       134036       437140        20088       377104       736824  
Swap:          102396           0        102396
```




네트워크 관리 명령어

- 호스트명 출력
 - **hostname**
- 텔넷 접속
 - **telnet** [IP주소]
- ftp접속
 - **ftp** [IP주소 또는 호스트 이름]
 - 사용 예) ftp>get [파일명]
- 네트워크 확인 명령
 - **ping** [옵션] [IP주소 또는 호스트 이름]



- LAN 카드 상태 확인
 - **ifconfig** [인터페이스] [주소] [up/down] [옵션]
- 네트워크 상태 확인
 - **netstat** [옵션] [address family options]
- `$sudo iwlist wlan0 scan` : 무선랜에 연결할 공유기를 찾는 명령어



■ vi 에디터 명령어

명령어		설명
시작	vi file	Vi를 시작하며 지정한 파일 편집
종료	:wq	데이터를 저장하고 종료
편집	:q!	데이터를 저장하지 않고 종료
커서이동	h i k l	커서를 한 칸 왼쪽으로 이동 커서를 한 줄 아래로 이동 커서를 한 줄 위로 이동 커서를 한 칸 오른쪽으로 이동
문자열치환	:s/pattern/replace/ :lines/pattern/replace/ :line, lines/pattern/replace/ :%s/pattern/replace/	현재 줄에서 pattern을 replace로 바꿈 lines줄의 pattern을 replace로 바꿈 line에서 lines까지의 줄에서 pattern을 replace로 바꿈 파일 전체에서 pattern을 replace로 바꿈
삽입	i a	입력 모드로 전환, 커서 위치 앞에서 삽입 입력 모드로 전환, 커서 위치 뒤에서 삽입
삭제	x dd	커서가 있는 위치의 문자 삭제 현재 줄을 전체 삭제



■ nano 에디터 명령어

명령어		설명
저장	Ctrl+O	바뀐 내용 저장
나가기	Ctrl+X	nano를 끝내기, 저장이 되지 않는다면 저장할지를 묻게 됨, 저장하지 않으려면 'N', 저장하고 싶으면 'Y' 새로 파일을 만든 경우, 미리 이름을 주지 않았다면 파일명을 물음, 파일명을 입력하고 Enter를 누름
취소	Ctrl+C	편집내용 취소
한 줄 잘라내기	Ctrl+K	Ctrl키와 K를 동시에 누르면 줄이 사라짐
붙여넣기	Ctrl+U	한 줄 잘라낸 내용은 ^U(Ctrl+U)로 붙여넣기 기능
문자열 찾기	Ctrl+w	^W(Ctrl+W)를 사용



정리하기

- 리눅스의 기본 명령어가 있는 디렉터리는?
- /dev, /ect, /sbin은 각각 어떤 디렉터리인가?
- file1을 file2에 복사하면서 겹쳐 쓸 것을 확인하는 명령어는?
- file1을 dir2라는 디렉터리로 옮기는 명령어는?
- file1 파일을 최상위 루트에서부터 동일한 파일을 검색하는 명령어는?
- chmod 755 my.py의 뜻은?
- IP 상태 및 랜카드 상태 확인하는 명령어는?
- 네트워크 상태 확인하는 명령어는?

Raspberry Pi를 활용한 IoT 프로젝트

Thank You