

# 리눅스에 대한 이해

- 파일 시스템
- 리눅스 명령어 기초
- 커맨드라인에서 파일 편집하기



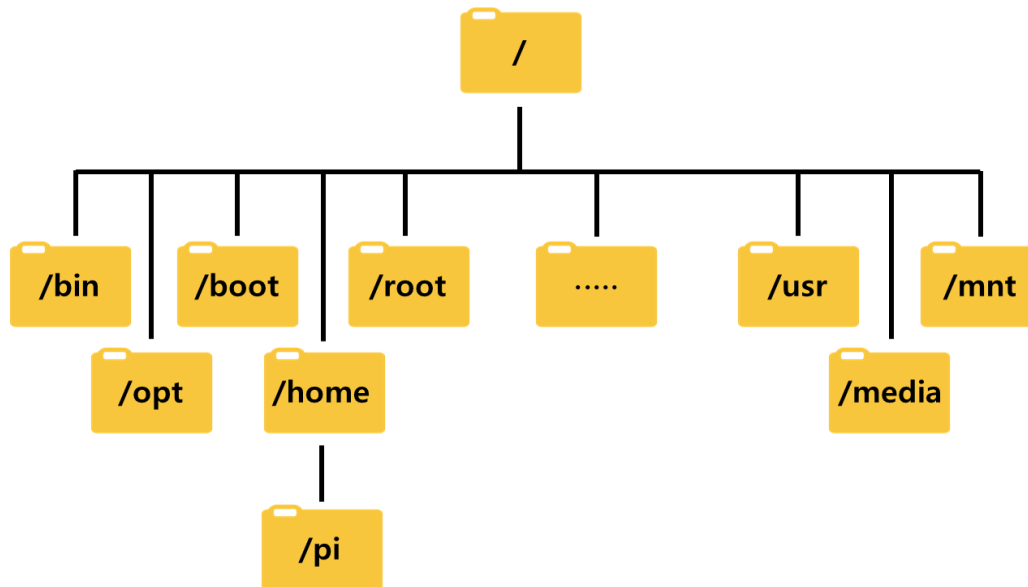
엣지아이랩

---

# 파일 시스템

# 파일 시스템

- 파일시스템이란?
  - 컴퓨터에서 파일이나 자료를 쉽게 발견하거나 접근할 수 있도록 보관 또는 조직하는 체계
  - 운영체제들은 내부에 단 하나의 파일 시스템만 지원
- 통합형 파일 시스템
  - 하나의 루트 디렉터리아래 운영체제에 접근할 수 있는 파일들은 루트 디렉터리 안에 있는 형태
  - 루트 디렉터리는 '/'로 표시
  - 일반 사용자와 관리자(root) 계정 존재



# 파일 시스템

- 파일 시스템 배치도

```
File Edit Tabs Help
pi@raspberrypi:~$ ls -a
.  bin  dev  home  lost+found  media  opt  root  sbin  sys  usr
.. boot etc  lib  man       mnt   proc  run  srv  tmp  var
pi@raspberrypi:~$
```

- / (루트 디렉터리)
  - 최상위 계층의 파일
- /root
  - 루트 사용자의 홈 디렉터리
  - 루트 디렉터리(/)와 혼동 조심
- /etc
  - 시스템 및 설치 어플리케이션의 환경 설정 파일 저장
- /proc
  - 커널이 사용자 영역의 도구들로부터 편리한 액세스를 제공하기 위한 가상 파일 시스템
  - 시스템 상태나 실행중인 프로세스 등에 관한 정보 포함

# 파일 시스템

- 파일 시스템 배치도
  - /var
    - 시스템이나 어플리케이션이 만든 파일 저장
  - /boot
    - 부트로더나 리눅스 커널 등 시스템 부팅에 필요한 주요 파일 저장
  - /bin & /sbin
    - 각각 사용자 프로그램과 관리용 프로그램 저장
  - /dev
    - 장치에 해당하는 파일 저장
    - 시스템 도구가 특정 컴퓨터의 하드웨어에 간편한 방법으로 접근 가능하게 함
  - /home
    - 사용자의 홈 디렉터리
  - /lib
    - 어플리케이션들에 필요한 라이브러리 파일 저장
    - 라이브러리는 패키지 형태로 공유

# 파일 시스템

- 파일 시스템 배치도
  - /lost+found
    - 디스크의 오류나 부적절한 시스템 종료에 의해 발견된 결함 파일에 대한 정보 보관
  - /mnt
    - Mount의 약자로 파일시스템을 추가로 마운트 하는 곳
  - /media
    - USB 스틱이나 카메라, 미디어 플레이어 등 이동식 미디어 마운트
    - 리눅스가 자동으로 관리
  - /usr
    - 컴퓨터에서 다수의 소프트웨어가 머무르는 곳
  - /opt
    - 프로그램을 추가로 설치할 경우에 사용하는 디렉터리
  - /srv
    - 시스템이 제공하는 서비스를 위한 파일 저장
  - /sys
    - 시스템 정보 보관
  - /tmp
    - 임시 파일 저장소

---

# 리눅스 명령어 기초

# 리눅스 명령어 기초

- 사용자 계정 관련 명령어
  - 사용자 계정이란?
    - 리눅스 시스템에서 컴퓨터 사용을 위해 관리자로부터 발급받는 사용자 인증을 위한 Login Name과 동시에 생성되는 디스크 상의 작업공간
    - 라즈베리파이의 경우
      - 기본 사용자 계정 'pi'
      - 관리자 계정 'root'
  - Sudo
    - 단독사용이 아닌 관리자(root)권한이 필요한 명령을 실행할 경우, 명령어 앞에 덧붙여 사용
  - passwd
    - 사용자의 login password 변경에 사용
    - 단독 사용할 경우, 현재 login 계정의 password변경
    - 'sudo'와 함께 사용시 다른 계정의 비밀번호 설정 가능
    - 터미널 창에 "sudo passwd root" 입력시 관리자 계정의 비밀번호 설정

```
File Edit Tabs Help
pi@raspberrypi:~ $ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~ $
```



# 리눅스 명령어 기초

- 사용자 계정 관련 명령어

- who

- 현재 자신이 login 되어있는 시스템에 함께 로그인 되어있는 사용자 확인
    - Login한 터미널의 이름은 login을 요구하는 시스템 메시지에 'tty1' 표시
    - ctrl+alt+F2를 누르면 tty2 콘솔 열기 가능(F1~F6 6개의 터미널 확장 가능)

```
File Edit Tabs Help
pi@raspberrypi:~ $ who
pi      :0                2017-07-31 09:12 (:0)
pi      tty1             2017-07-31 09:12
pi@raspberrypi:~ $
```

- whoami

- 사용자 자신이 login한 login name이 무엇인지 알아보는 명령어

```
File Edit Tabs Help
pi@raspberrypi:~ $ whoami
pi
pi@raspberrypi:~ $
```

# 리눅스 명령어 기초

- 사용자 계정 관련 명령어

- useradd

- 사용자 계정 추가 명령어
    - 명령 뒤에 옵션 및 계정 ID를 붙여 사용
    - "useradd [옵션] [계정 ID]"
      - -u : 입력한 UID 로 계정을 추가
      - -g : 입력한 GID 로 계정을 추가
      - -d : 입력한 경로로 홈디렉토리를 지정 후 계정을 추가

```
File Edit Tabs Help
pi@raspberrypi:~ $ sudo useradd iot
pi@raspberrypi:~ $ sudo passwd iot
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~ $
```

- userdel

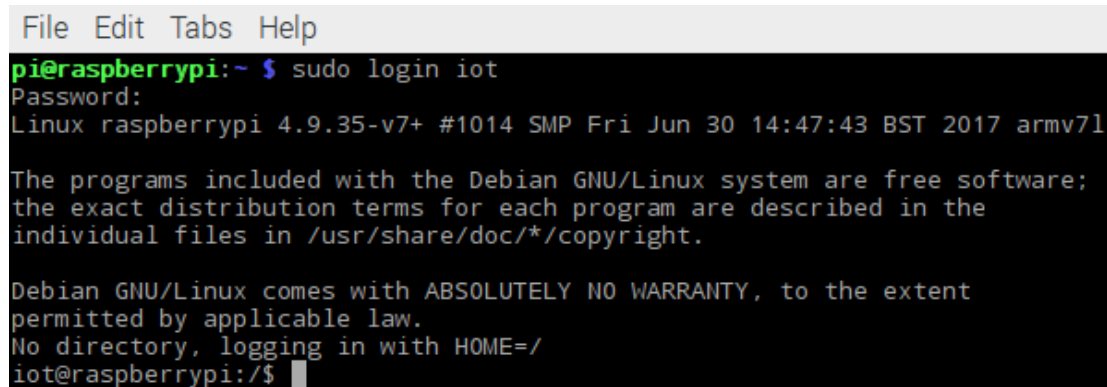
- 사용자 계정 삭제 명령어
    - "userdel [옵션] [계정 ID]"
      - -r : 계정과 계정의 홈 폴더도 함께 삭제

# 리눅스 명령어 기초

- 사용자 계정 관련 명령어

- login

- 사용자 계정으로 로그인하기 위한 명령어



```
File Edit Tabs Help
pi@raspberrypi:~ $ sudo login iot
Password:
Linux raspberrypi 4.9.35-v7+ #1014 SMP Fri Jun 30 14:47:43 BST 2017 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
No directory, logging in with HOME=/
iot@raspberrypi:/$
```

- logout

- 로그인 과정에서 빠져 나오는 명령어
    - PuTTY를 통한 원격 로그인의 경우 "exit"로 대신함

# 리눅스 명령어 기초

- 파일 및 디렉터리 관련 명령어

- 디렉터리(directory)란?

- MS-Windows에서의 폴더(folder)와 동일한 개념
    - 어떤 기준에 의해 파일(file)들을 모아둔 파일 시스템 상의 논리적 공간

- pwd(path working directory)

- 현재 사용자의 디렉터리(경로)를 알려주는 명령어

```
File Edit Tabs Help
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $
```

- ls(list)

- 단독 사용의 경우, 현재 위치의 파일 및 디렉터리 목록을 출력
    - 뒤에 특정 디렉터리 위치를 적어줄 경우, 해당 디렉터리 파일 및 디렉터리 목록을 출력

```
File Edit Tabs Help
pi@raspberrypi:~ $ ls
Desktop  Downloads  Pictures  python_games  test.txt
Documents Music      Public    Templates      Videos
pi@raspberrypi:~ $
```

# 리눅스 명령어 기초

- 파일 및 디렉터리 관련 명령어

- ls(list)

- "ls [옵션] [디렉터리/파일]"

- -a : 숨겨져 있던 파일을 포함하여 모두 출력

```
File Edit Tabs Help
pi@raspberrypi:~ $ ls -a
.          .config    .gconf     .profile   .thumbnails
..         .cups      .gvfs      Public     Videos
.bash_history .dbus     .local     python_games .vnc
.bash_logout Desktop    Music      Templates  .Xauthority
.bashrc     Documents Pictures    test.txt   .xsession-errors
.cache      Downloads .pki       .themes    .xsession-errors.old
pi@raspberrypi:~ $
```

- -l : 파일에 관한 상세정보를 리스트(list) 형태로 출력

```
File Edit Tabs Help
pi@raspberrypi:~ $ ls -l
total 40
drwxr-xr-x 2 pi pi 4096 Jul  5 12:01 Desktop
drwxr-xr-x 5 pi pi 4096 Jul  5 11:29 Documents
drwxr-xr-x 2 pi pi 4096 Jul  5 12:01 Downloads
drwxr-xr-x 2 pi pi 4096 Jul  5 12:01 Music
drwxr-xr-x 2 pi pi 4096 Jul  5 12:01 Pictures
drwxr-xr-x 2 pi pi 4096 Jul  5 12:01 Public
drwxr-xr-x 2 pi pi 4096 Jul  5 11:29 python_games
drwxr-xr-x 2 pi pi 4096 Jul  5 12:01 Templates
-rw-r--r-- 1 pi pi  20 Jul 28 01:02 test.txt
drwxr-xr-x 2 pi pi 4096 Jul  5 12:01 Videos
pi@raspberrypi:~ $
```

# 리눅스 명령어 기초

- 파일 및 디렉터리 관련 명령어
  - ls(list)
    - "ls [옵션] [디렉터리/파일]"
      - -F : 파일 유형을 나타내는 기호를 파일명 끝에 표시
        - 디렉터리는 '/', 실행파일은 '\*' 표시

```
File Edit Tabs Help
pi@raspberrypi:~ $ ls -F
Desktop/  Downloads/  Pictures/  python_games/  test.txt
Documents/  Music/      Public/    Templates/      Videos/
pi@raspberrypi:~ $
```

- -R : 해당 디렉터리의 서브 디렉터리까지 모두 출력

```
File Edit Tabs Help
pi@raspberrypi:~ $ ls -R
.:
Desktop  Downloads  Pictures  python_games  test.txt
Documents  Music      Public    Templates      Videos

./Desktop:

./Documents:
BlueJ Projects  Greenfoot Projects  Scratch Projects

./Documents/BlueJ Projects:
appletdemo  hello  people  README.TXT
debugdemo   LED-Button  people2  shapes
file-reader  LICENSE.txt  picture  THIRDPARTYLICENSE.txt
```

# 리눅스 명령어 기초

- 파일 및 디렉터리 관련 명령어

- mkdir(make directory)

- 새로운 디렉터리 생성

- "mkdir [옵션] [이름]"

- -m : 디렉터리 생성시 기본권한 설정

- -p : 상위 디렉터리 생성

```
pi@raspberrypi:~ $ mkdir EDGEILAB
pi@raspberrypi:~ $ ls
Desktop  Downloads  Music      Public      Templates  Videos
Documents EDGEILAB    Pictures   python_games test.txt
```

- cd(change directory)

- 현재의 디렉터리 이동

- "cd [디렉터리/인자값]"

- cd . : 현재 디렉터리로 이동

- cd .. : 상위 디렉터리로 이동

- Cd ~ : 홈 디렉터리로 이동

- Cd - : 이전 작업 디렉터리로 이동

```
pi@raspberrypi:~ $ cd EDGEILAB/
pi@raspberrypi:~/EDGEILAB $ cd ..
pi@raspberrypi:~ $
```

# 리눅스 명령어 기초

- 파일 및 디렉터리 관련 명령어

- touch

- 파일을 생성해주거나, 생성된 파일의 경우 시간정보를 변경함
    - "touch [옵션] [파일이름]"
      - -r : 시간 동기화
      - -t : 지정 시간으로 변경

```
pi@raspberrypi:~/EDGEILAB $ touch test.txt
pi@raspberrypi:~/EDGEILAB $ ls
test.txt
pi@raspberrypi:~/EDGEILAB $
```

- cp(copy)

- 해당 파일을 복사
    - "cp [옵션] [복사파일] [복사위치]"
      - -f : 강제로 복사
      - -r : 하위 경로 포함하여 복사
      - -v : 복사 진행상황 출력

```
pi@raspberrypi:~/EDGEILAB $ ls
test.txt
pi@raspberrypi:~/EDGEILAB $ cp test.txt ../
pi@raspberrypi:~/EDGEILAB $ cd ..
pi@raspberrypi:~ $ ls
Desktop  Downloads  Music      Public      Templates  Videos
Documents EDGEILAB   Pictures   python_games test.txt
```



# 리눅스 명령어 기초

- 파일 및 디렉터리 관련 명령어

- mv(move)

- 해당 파일을 이동하거나 파일명 변경에 쓰는 명령어
    - "mv [옵션] [이동할 파일] [디렉터리/파일]"
      - -i : 이동에 대한 실행 여부 확인
      - -f : 강제로 이동
      - -v : 이동 진행 상황 출력

```
pi@raspberrypi:~/EDGEILAB $ ls
test.txt
pi@raspberrypi:~/EDGEILAB $ mv test.txt test1.txt
pi@raspberrypi:~/EDGEILAB $ ls
test1.txt
```

- rm(remove)

- 해당 파일 및 디렉터리 삭제
    - "rm [옵션] [디렉터리/파일]"
      - -f : 강제 삭제
      - -r : 디렉터리 삭제시 하위경로 포함하여 삭제
      - -v : 삭제 진행 상황 출력

```
pi@raspberrypi:~/EDGEILAB $ ls
test1.txt
pi@raspberrypi:~/EDGEILAB $ rm test1.txt
pi@raspberrypi:~/EDGEILAB $ ls
pi@raspberrypi:~/EDGEILAB $
```

# 리눅스 명령어 기초

- 파일 및 디렉터리 관련 명령어
  - chmod
    - 파일의 속성 변경

```
File Edit Tabs Help
pi@raspberrypi:~/EDGEILAB $ ls -l
total 4
-rw-r--r-- 1 pi pi 16 Aug  1 02:11 test.txt
pi@raspberrypi:~/EDGEILAB $ chmod 555 test.txt
pi@raspberrypi:~/EDGEILAB $ ls -l
total 4
-r-xr-xr-x 1 pi pi 16 Aug  1 02:11 test.txt
pi@raspberrypi:~/EDGEILAB $
```

- cat
  - 텍스트 파일의 내용을 화면에 그대로 출력
  - "cat [옵션] [파일이름]"
    - > : 내용 덮어 씌우기
    - >> : 기존 파일 내용 추가

```
File Edit Tabs Help
pi@raspberrypi:~/EDGEILAB $ cat test.txt
HELLO WORLD!!!!
pi@raspberrypi:~/EDGEILAB $
```

# 리눅스 명령어 기초

- 파일 및 디렉터리 관련 명령어
  - find
    - 해당파일 검색
    - "find [디렉터리] [옵션] [파일명]"
      - -name : 같은 단어가 들어간 파일을 검색
      - Ex) "/home/pi" 디렉터리에서 모든 txt파일 검색

```
File Edit Tabs Help
pi@raspberrypi:~ $ find /home/pi -name *.txt
/home/pi/Downloads/test.txt
/home/pi/test.txt
/home/pi/EDGEILAB/test.txt
/home/pi/Documents/BlueJ Projects/file-reader/test.txt
pi@raspberrypi:~ $
```

# 리눅스 명령어 기초

- 파일의 속성
  - 리눅스는 다수의 사용자들이 동시에 시스템에 접근 가능
  - 파일의 소유자는 서로 다를 수 있음
  - 보안을 위해 각 파일에 대해 사용자 별로 읽기, 쓰기, 실행 권한 부여 필요
  - 터미널 창에 "ls -al" 입력
    - 설정된 권한 확인

```
File Edit Tabs Help
pi@raspberrypi:~ $ ls -al
total 120
drwxr-xr-x 23 pi pi 4096 Aug  1 02:08 .
drwxr-xr-x  3 root root 4096 Jul  5 10:41 ..
-rw----- 1 pi pi 3824 Jul 31 09:12 .bash_history
-rw-r--r-- 1 pi pi  220 Jul  5 10:41 .bash_logout
-rw-r--r-- 1 pi pi 3512 Jul  5 10:41 .bashrc
drwxr-xr-x  7 pi pi 4096 Jul 25 05:15 .cache
drwx----- 14 pi pi 4096 Jul 31 02:07 .config
drwx-----  2 pi pi 4096 Jul 18 08:36 .cups
drwx-----  3 pi pi 4096 Jul 18 08:09 .dbus
drwxr-xr-x  2 pi pi 4096 Jul  5 12:01 Desktop
drwxr-xr-x  5 pi pi 4096 Jul  5 11:29 Documents
drwxr-xr-x  2 pi pi 4096 Aug  1 01:37 Downloads
drwxr-xr-x  2 pi pi 4096 Aug  1 02:11 EDGEILAB
drwx-----  2 pi pi 4096 Jul 25 05:58 .gconf
```

# 리눅스 명령어 기초

- 파일의 속성
  - "drwxr-xr-x"와 같이 사용자 권한 명시

```
pi@raspberrypi:~ $ ls -al
total 120
drwxr-xr-x 23 pi   pi   4096 Aug  1 02:08 .
drwxr-xr-x  3 root root 4096 Jul  5 10:41 ..
```

- 10개의 문자, 4블록 구성

유형	소유자			그룹			그외 사용자		
	r	w	x	r	w	x	r	w	x
	r(읽기), w(쓰기), x(실행)								

- Ex) 파일 속성이 "-rwxr-xr-x" 인 경우

유형	소유자			그룹			그외 사용자		
일반 파일	읽기	쓰기	실행	읽기	쓰기	실행	읽기	쓰기	실행
	○	○	○	○		○	○		○

# 리눅스 명령어 기초

- 파일의 유형
  - 맨 앞의 첫번째 문자는 파일의 유형
  - 대부분 'd'(디렉터리) 혹은 '-'(일반 파일)로 시작

파일 유형	내 용
d	디렉터리
-	일반 파일
l	링크 파일
c	캐릭터 형태의 장치 파일
b	블록 형태의 장치 파일
s	소켓 파일
p	파이프 파일

# 리눅스 명령어 기초

## ● 파일의 사용권한

- 소유자, 그룹, 일반 사용자 세 분류
- 10개 문자 중, 첫 문자를 제외한 나머지 9문자로 구성
- 첫 3자리는 파일 소유자의 권한
- 그 뒤 3자리는 소유자가 속한 그룹에 대한 권한
- 마지막 3자리는 나머지 일반 사용자들의 권한 명시
- 문자 대신 숫자를 사용하는 경우도 있음
- Ex) 파일 속성이 "-rwxr-xr--"인 경우
  - - : 일반파일
  - rwx : 파일 소유자는 읽기, 쓰기, 실행 가능
  - r-x : 그룹은 읽기, 실행 가능
  - r-- : 일반 사용자는 읽기만 가능
- 리눅스 명령어 "chmod"를 통해 특정 파일 권한 변경 가능

소유자(user)			그룹(group)			일반사용자(other)		
r	w	x	r	-	x	r	-	-
4	2	1	4	0	1	4	0	0
7			5			4		

# 커맨드라인에서 파일 편집하기



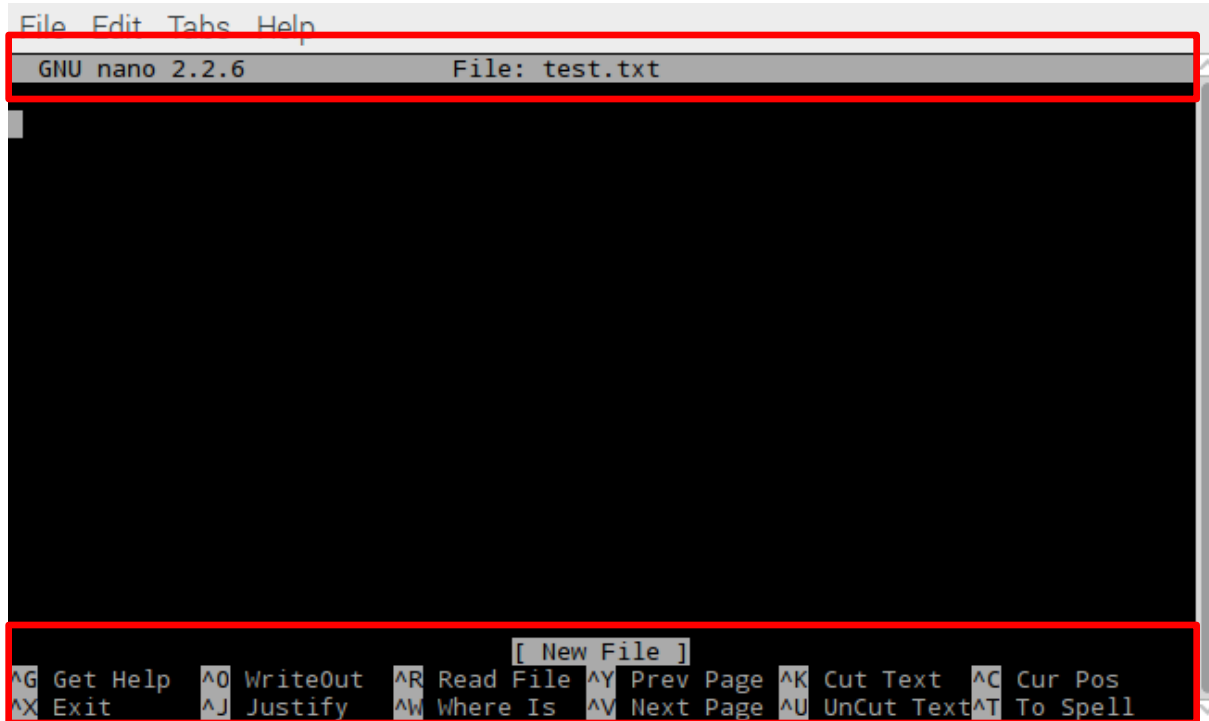
# 커맨드라인에서 파일 편집하기

- 텍스트 파일이란?
  - 나노(nano)
    - UNIX 계열 컴퓨팅 시스템이나 명령줄 인터페이스를 사용하는 운영 환경을 위한 초경량 텍스트 에디터
    - 사용법이 쉬움
  - VI(Visual Editor)
    - UNIX계열 운영체제에서 주로 쓰이는 오픈소스 텍스트 에디터
    - 편집 자체를 따로 하나의 모드로 구성
    - 키조작이 단순
    - 입력, 명령, 비주얼 등의 모드가 존재
    - 일반적으로 기능을 모방하여 만들어진 클론(Vim) 사용

# 커맨드라인에서 파일 편집하기

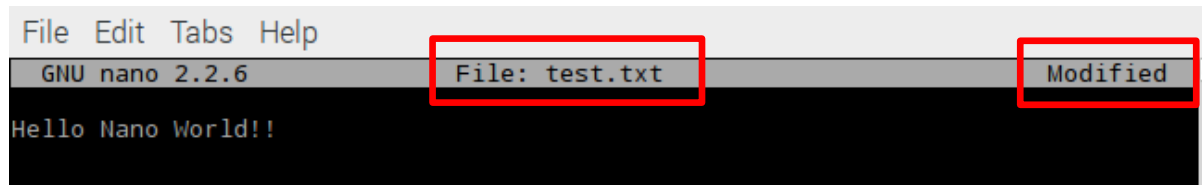
- 나노 시작하기
  - 터미널 창에 "nano test.txt" 입력
    - 나노를 사용하여 test.txt 파일 열기
    - 해당 파일이 존재하지 않을 경우 새로 생성

```
pi@raspberrypi:~ $ nano test.txt
```



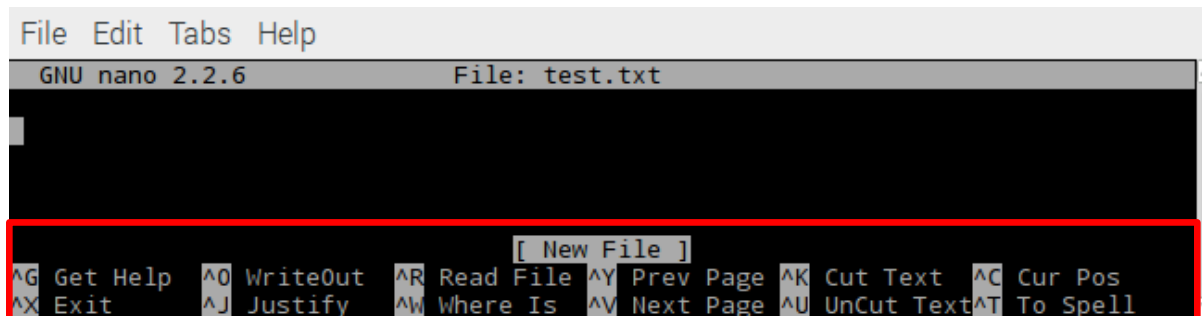
# 커맨드라인에서 파일 편집하기

- 나노(nano) 시작하기
  - 제목 표시줄에 현재 파일 이름 확인
  - 수정된 경우, 우측 상단에 "Modified" 표시



```
File Edit Tabs Help
GNU nano 2.2.6 File: test.txt Modified
Hello Nano World!!
```

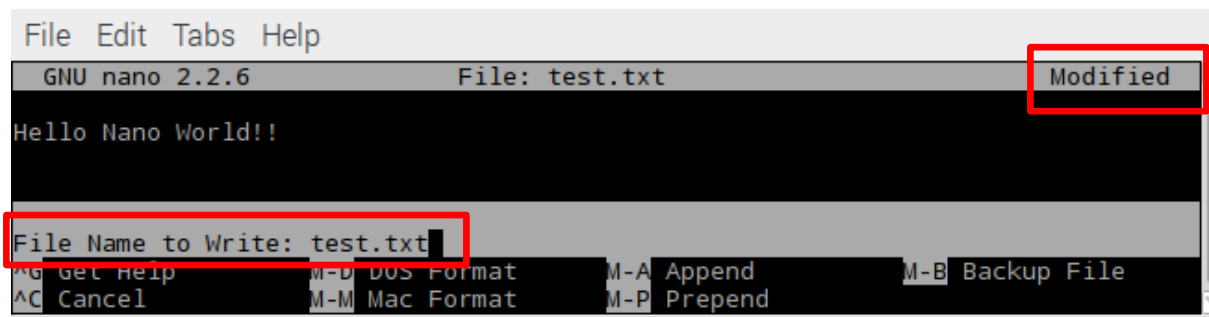
- 화면 하단에 단축키와 메뉴 표시줄 확인
  - 옵션 선택 : ctrl(^)+[옵션]



```
File Edit Tabs Help
GNU nano 2.2.6 File: test.txt
[ New File ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

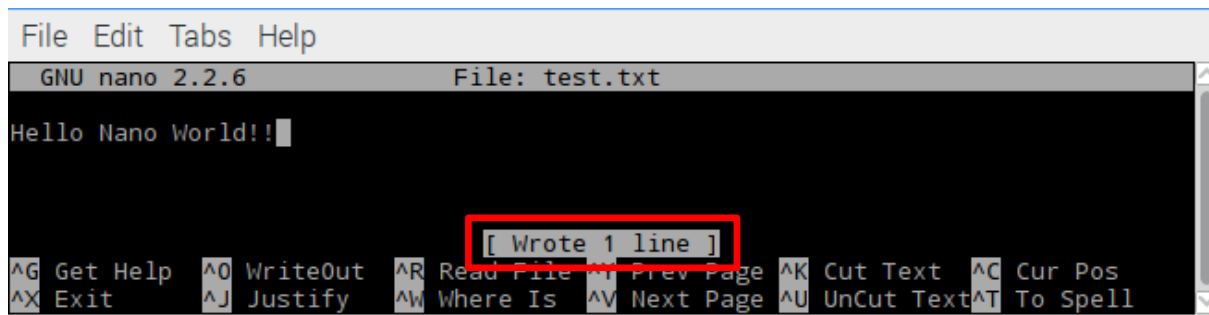
# 커맨드라인에서 파일 편집하기

- 나노(nano) 사용하기
  - 파일 저장하기(WriteOut)
    - 단축키 “^O” 사용
    - 하단에 저장할 파일명 확인



The screenshot shows the GNU nano 2.2.6 editor interface. The top status bar displays 'GNU nano 2.2.6' and 'File: test.txt'. The main text area contains 'Hello Nano World!!'. A red box highlights the 'File Name to Write: test.txt' prompt at the bottom of the editor. The bottom status bar shows various keyboard shortcuts, including '^O' for WriteOut.

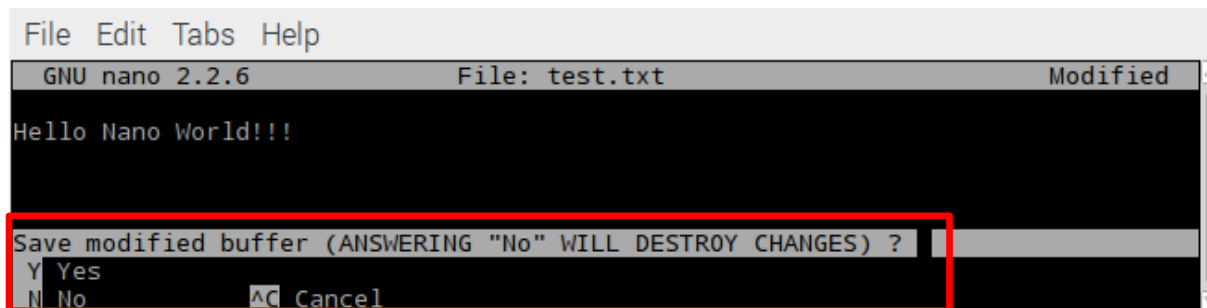
- 저장 완료 시, “Modified”가 사라지고 1 line을 저장했음을 확인



The screenshot shows the GNU nano 2.2.6 editor interface after saving. The top status bar displays 'GNU nano 2.2.6' and 'File: test.txt'. The main text area contains 'Hello Nano World!!'. A red box highlights the '[ Wrote 1 line ]' message at the bottom of the editor. The bottom status bar shows various keyboard shortcuts, including '^O' for WriteOut.

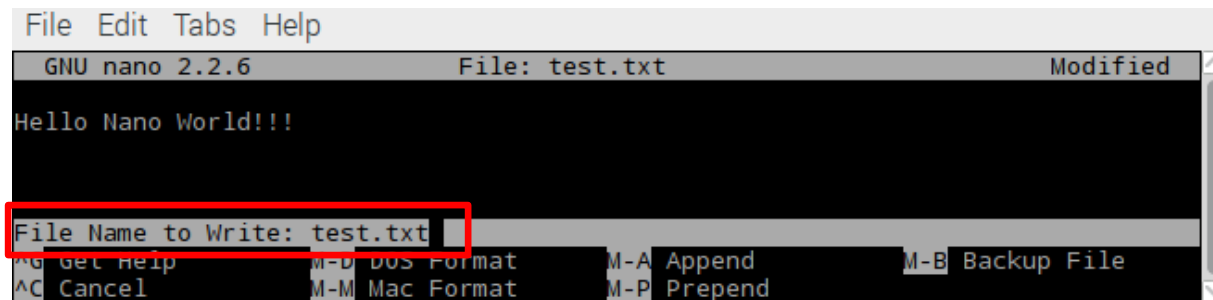
# 커맨드라인에서 파일 편집하기

- 나노(nano) 사용하기
  - 종료하기
    - 단축키 "**^X**" 사용
    - 저장한 경우, 바로 종료
    - 저장을 안한 경우, 저장 할 것인지 묻고 종료
      - 저장 : "**Y**"
      - 저장안함 : "**N**"



The screenshot shows the GNU nano 2.2.6 editor interface. The menu bar at the top includes 'File', 'Edit', 'Tabs', and 'Help'. The status bar below the menu shows 'GNU nano 2.2.6', 'File: test.txt', and 'Modified'. The main text area contains 'Hello Nano World!!!'. A red box highlights the prompt 'Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?' with options 'Y Yes' and 'N No', and a keyboard shortcut '^C Cancel'.

- 저장할 경우 파일명 확인



The screenshot shows the GNU nano 2.2.6 editor interface. The menu bar at the top includes 'File', 'Edit', 'Tabs', and 'Help'. The status bar below the menu shows 'GNU nano 2.2.6', 'File: test.txt', and 'Modified'. The main text area contains 'Hello Nano World!!!'. A red box highlights the prompt 'File Name to Write: test.txt'. Below the prompt, a keyboard shortcut '^C Cancel' is visible.