

얼떨결에 사용자 분들의 학교 홈페이지 서버 이용을 안내하게 된 허접가이드 lkach 입니다..

여기서 허접 가이드 lkach 가 안내하는 것은 아주 기본적인 사용 방법입니다. (잘 아시는 분은 볼 필요 없습니다..

결정적으로 lkach 도 이제서야 유닉스를 접하게된 허접이라서 잘 모릅니다. ㅠ ㅠ) 때문에 아주 자세히 알고 싶으신 분은 학교 도서관에서 유닉스 책을 빌려 보시거나 사서 보시면 되겠습니다. (참고: 일단 학교 홈페이지 학부생 서버 iam 서버를 기준으로 했습니다.)

(1) 서버에 접속하기 입니다.

학부생 서버인 iam.snut.ac.kr 서버에 접속해 보겠습니다.

윈도우즈 98 등을 이용하시는 분은 MS-DOS 창을 이용하시거나 시작-> 실행 -> command 라고 입력 하시면 도스창 처럼 생긴 화면이 모니터에 나타 납니다. 여기서 telnet iam.snut.ac.kr 이라고 입력하시면

```
#
#
# WARNING: You must have specific authorization to access or
#           use this system. All connections to this system
#           are logged and monitored. Unauthorized access or
#           use will be prosecuted.
#
login:
```

이런 화면이 나타납니다.

여기에 여러분이 받으신 홈페이지 계정 아이디를 입력하시고 패스워드를 입력하시면 됩니다.

여건상 제 아이디인 lkach 를 넣어 보겠습니다. (참고: 여기 에서는 delete 키로만 삭제가 됩니다. 뒤로 가는 화살표 모양의 키 즉 백 스페이스 키로는 지워지지 않습니다. 이상한 기호가 입력되실 겁니다. ^H 이런 모양이지요..)

실제로 접속해서 아이디와 패스워드를 입력한 후의 모습 입니다.

```
#
#
# WARNING: You must have specific authorization to access or
#           use this system. All connections to this system
#           are logged and monitored. Unauthorized access or
#           use will be prosecuted.
#
login: lkach
Password:
Last login: Mon May 27 21:29:39 from snut0041.snut.ac
Sun Microsystems Inc. SunOS 5.6 Generic August 1997
$
```

패스워드는 입력하는 동안에도 보이지 않습니다. 보안상 안보이게 했다고 하네요... 제가 그렇게 일부러 만든게 아닙니다. 윈도우 처럼 **** 식으로 나타나면 좋을 텐데 라는 생각이 들지만 으음. .쓸데 없는 이야기였습니다. 일단 학교 서버에 접속이 되신 겁니다.. 축하합니다. 저는 이걸 하는데 3년 걸렸습니다.

여러분은 학교 홈페이지서버에 주어진 여러분의 공간에 들어 오신 겁니다. (참고: iam 서버는 학부생에게 20 메가의 공간을 드리고 있습니다.)

그럼 일단 여러분의 공간이 서버의 어디쯤에 위치 하는 지 알아 봐야 겠죠.. 이럴땐 pwd 명령어를 입력하시면 됩니다.)

(2) 현재 자신의 위치를 확인해 보는 pwd 명령 입니다.

```
$ pwd
/home1/stud/lkach
$
```

pwd 명령은 절대 경로를 알려 줍니다. 즉 시스템상에 여러분의 작업 디렉토리가 어느곳에 위치하는지 알려 주는 명령어 입니다. 유닉스에서 디렉토리를 여기 저기 다니다 보면 자신이 어디쯤 와있는지 잊어버릴 경우가 있습니다. 그럴경우 나침반처럼 쓰시면 되는 명령어 입니다.

(참고 : 혹시 디렉토리가 무슨 의미인지 모르시는 분은... - - ;; 없으시리라 믿겠습니다. 도스를 써보셨다면 당연히 아실 겁니다. 윈도우즈 밖에 안 써보셨다면 윈도우즈의 탐색기를 떠올리시면 됩니다.)

이제 여러분의 디렉토리안에 무엇이 있는지 보겠습니다.

명령어는 ls 입니다 (참고 : l은 영문 소문자 엘 L 입니다. 대문자 안됩니다.)

(3) 디렉토리안의 내용을 보여주는 ls 명령에 대해 잠시 알아 보겠습니다.

도스의 dir 명령과 비슷 합니다.

```
$ ls
local.cshrc  local.login  local.profile  www_home
$
```

뭔가 4 개가 보이시지요.. 그냥 ls 명령을 입력하시면 파일 인지 디렉토리 인지 알아보기가 힘듭니다. 게다가 유닉스는 확장자 개념이 없습니다. 즉 윈도우즈 처럼 뭐뭐.exe 하면 실행화일이라는 식이 아닙니다. 이름이 자기 맘대로 입니다. 즉 윈도우즈는 test.txt 나 test.hwp 등등 파일이름위의 확장자 .txt 나 .hwp 등을 보면 문서 화일이구나 알수 있지만 유닉스에서는 알 수 없습니다. 예를 들어 testfile.test.test.test 라는 이상한 이름으로도 유닉스에서는 파일을 만들수가 있습니다.때문에 ls 명령에 옵션을 줘야 조금 알아 볼 수 있습니다.

가장 잘 쓰는 옵션은 -l (참고로 영문자 소문자 엘 L 입니다)옵션 입니다.

(참고: file 명령이나 기타 다른 방법으로도 자신이 가지고 있는 파일의 종류나 내용을 볼 수 있는 방법은 많습니다. 역시 자세히 알고 싶으시면 책을 빌려보세요)

```
$ ls -l
total 8
```

```
-rw-r--r--  1 lkach  stud      124  5월 13일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud      575  5월 13일  22:20 local.login
-rw-r--r--  1 lkach  stud      560  5월 13일  22:20 local.profile
drwxr-xr-x  3 lkach  stud      512  5월 27일  21:34 www_home
```

이제 조금 알아 보실 수 있을 겁니다. 그래도 모르시는 분을 위해서 설명을 드리겠습니다. drwxr-xr-x 이부분이 있지 않습니까.. drwxr 에서 맨앞의 d 는 디렉토리 입니다. 그리고 나머지 rw-r--r--이런 부분은 뒤의 권한 설정 부분에서 설명 드리겠습니다. 일단 디렉토리인지 아닌지 아는게 기본입니다. 다른 방법으로는 -CF 옵션을 쓰는 방법도 있습니다.(참고로 방법은 많습니다. 역시 자세히 알고 싶으시면 책을 빌려보세요)

```
$ ls -CF
local.cshrc    local.login    local.profile  www_home/
$
```

/부분이 있으면 디렉토리 입니다. 즉 www_home 은 디렉토리 입니다. 기타 많은 옵션이 있지만 옵션에 대한 자세한 사항은 위에서 말씀것 처럼 유닉스 책을 참조하시길 바랍니다. ls 명령어 의 옵션도 꽤 많습니다... 이거 설명하려면..음.. 그냥 여기서는 기본 옵션만 알고 쓰시다.

(4) 빈 화일 생성 하기 입니다. touch 명령을 이용합니다.

```
$ touch testfile <-testfile 이라는 이름으로 파일을 만들어 봤습니다.
$ ls -l
total 8
-rw-r--r--  1 lkach  stud      124  5월 13일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud      575  5월 13일  22:20 local.login
-rw-r--r--  1 lkach  stud      560  5월 13일  22:20 local.profile
-rw-r--r--  1 lkach  stud         0  5월 31일  11:22 testfile
drwxr-xr-x  3 lkach  stud      512  5월 27일  21:34 www_home
$
```

touch 명령은 빈 화일을 만들어 주는 명령입니다. touch 명령 뒤에 임의의 화일명을 적고 입력하면 됩니다.여기에서는 testfile 이 만들어 저 있음을 확인 할 수 있습니다.

(5) 디렉토리 만들기 입니다.

```
$ mkdir testdir <- testdir 이라는 이름으로 디렉토리를 만들어 봤습니다.
$ ls -l
total 10
-rw-r--r--  1 lkach  stud      124  5월 13일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud      575  5월 13일  22:20 local.login
-rw-r--r--  1 lkach  stud      560  5월 13일  22:20 local.profile
```

```
drwxr-xr-x  2 lkach  stud    512  5월 31일  11:24 testdir
-rw-r--r--  1 lkach  stud      0  5월 31일  11:22 testfile
drwxr-xr-x  3 lkach  stud    512  5월 27일  21:34 www_home
$
```

디렉토리를 만드는 명령어 입니다. 입력 형식은 보시는 바와 같습니다. ls 명령으로 확인해 보면 testdir 이라는 임의의 디렉토리가 생성되어 있음을 확인 할 수 있습니다.

(6) 디렉토리를 이동하는 cd 명령입니다.

```
$ cd testdir
$ ls -l
total 0  <- ls 명령으로 살펴본 결과 testdir 디렉토리 안에는 아무런 파일도 없음을 확인 할 수
있습니다.
$ cd ..  <- cd 라고 입력하시고 한칸 띄우고 마침표 두개..를 적으시면 바로 상위
디렉토리로이동한다.
$ ls -l
total 10
-rw-r--r--  1 lkach  stud    124  5월 13일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud    575  5월 13일  22:20 local.login
-rw-r--r--  1 lkach  stud    560  5월 13일  22:20 local.profile
drwxr-xr-x  2 lkach  stud    512  5월 31일  11:24 testdir
-rw-r--r--  1 lkach  stud      0  5월 31일  11:22 testfile
drwxr-xr-x  3 lkach  stud    512  5월 27일  21:34 www_home
$
```

다시한번 ls 명령 으로 확인해본 결과 testdir 디렉토리에서 다시 원래의 디렉토리로 돌아와 있음을 확인 할 수 있습니다. 이외에도 위에서 나온 pwd 명령으로 경로를 지정해서 직접 이동하는 방법도 있습니다.

```
$ cd testdir  <- 다시 testdir 디렉토리로 이동하신후
$ pwd        <- pwd 명령으로 여기가 어디인지 확인해 봅니다.
/home1/stud/lkach/testdir
$ cd ..      <- 다시 cd .. 명령으로 상위 디렉토리로 이동한후
$ pwd        <- pwd 명령으로 여기가 어디인지 확인해 봅니다
/home1/stud/lkach                <- 현재 디렉토리 입니다.

$ cd /home1/stud/lkach/testdir  <- 이번에는 경로를 직접 지정해서 testdir 디렉토리로 이동해 봅니다.
$ pwd
/home1/stud/lkach/testdir        <- 제대로 온게 맞죠...
```

(7) 파일을 복사하는 방법입니다. cp 명령을 쓰시면 됩니다.

```
$ cp testfile testdir <- testdir 디렉토리에 testfile 을 복사해서 넣는 명령입니다.  
$ ls  
local.cshrc  local.profile  testfile  
local.login  testdir        www_home  
$ cd testdir <- testdir 디렉토리로 가서 복사가 되었는지 확인해 보면.  
$ ls -l  
total 0  
-rw-r--r--  1 lkach  stud          0  5월 31일  11:42 testfile <-  
testfile 라는 파일이 하나 생성되 있음을 보실 수 있을 겁니다.  
$
```

(8) 삭제 하는 방법입니다.

이번에는 삭제하는 방법 입니다. 복사를 했으면 삭제도 해야죠...^^;; rm 명령을 쓰시면 됩니다.

```
<- testdir 디렉토리 안에 복사했던 testfile 을  
$ rm testfile <- rm 명령으로 지우는 명령입니다.  
$ ls -l <- ls 명령으로 확인 해 보시면 지워진 것을 보실 수  
있으실  
<- 겁니다.  
total 0  
$  
$ cd .. <- 다시 원래의 디렉토리 (처음 로그인 했을때 들어갔던  
<- 자신의 홈 디렉토리 즉 여기서는 lkach 디렉토리를  
말합니다.)로  
<- 이동을 해봤습니다.  
$ ls -l  
total 10  
-rw-r--r--  1 lkach  stud          124  5월 13일  22:20 local.cshrc  
-rw-r--r--  1 lkach  stud          575  5월 13일  22:20 local.login  
-rw-r--r--  1 lkach  stud          560  5월 13일  22:20 local.profile  
drwxr-xr-x  2 lkach  stud          512  5월 31일  11:45 testdir  
-rw-r--r--  1 lkach  stud           0  5월 31일  11:42 testfile  
drwxr-xr-x  3 lkach  stud          512  5월 27일  21:34 www_home  
$
```

```

$ rm -r testdir      <- 이번에는 디렉토리를 삭제 하는 방법입니다. -r 옵션을
                    <- 주시면 다 지워집니다. 특히 -r 옵션을 쓰실때는
                    <- 주의를 하셔야 합니다. 진짜 다지워 집니다.
                    <- 디렉토리안에 들어있는 파일도 지워집니다. 잘못 쓰면
                    피웁니다.
$ ls -l
total 8
-rw-r--r--  1 lkach  stud      124  5 월 13 일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud      575  5 월 13 일  22:20 local.login
-rw-r--r--  1 lkach  stud      560  5 월 13 일  22:20 local.profile
-rw-r--r--  1 lkach  stud         0  5 월 31 일  11:42 testfile
drwxr-xr-x  3 lkach  stud      512  5 월 27 일  21:34 www_home
$                  <- ls 명령으로 디렉토리가 지워 졌음을 확인해 보았습니다.

```

(9) 권한 설정하기 입니다.

자신의 파일을 남들이 읽거나 쓰지 못하게 만들 수 있는 파일 권한 설정을 하시는 부분입니다.

```

$ ls -l
total 8
-rw-r--r--  1 lkach  stud      124  5 월 13 일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud      575  5 월 13 일  22:20 local.login
-rw-r--r--  1 lkach  stud      560  5 월 13 일  22:20 local.profile
-rw-r--r--  1 lkach  stud         0  5 월 31 일  11:42 testfile
drwxr-xr-x  3 lkach  stud      512  5 월 27 일  21:34 www_home

```

여기서 testfile 을 보시면 -rw-r--r--이라는 것을 보실 수 있습니다.

이 설정을 예로 들면 처음 한자리 ? 표시는 파일인지 디렉토리인지 등등의 종류를 나타냅니다. 그리고 rw- 와 r-- 와 r-- 의 세부분으로 나뉘서 . 첫번째 3 자리는 자신의 권한 , 두번째 3 자리는 그룹, 세번째 3 자리는 다른사람의 권한 설정 부분입니다. 그리고 그 각각의 3 자리는 읽기 쓰기 실행으로 나누어 집니다. -표시가 되었으면 그 권한이 없는것이고 r 이나(r 은 읽기 권한입니다.) w(w는 쓰기권한 입니다) 나 x(x 는 실행권한입니다) 로 되어있으면 일단 그 권한이 주어진 것이라고 생각하시면 됩니다. 위의 예에서 처음 한자리는 ? 이므로 디렉토리가 아닌 파일임을 알 수 있습니다. 그리고 처음 3 자리가 rw- 이므로 자신이 이 파일을 가지고 할 수 있는 일이 읽기와 쓰기 임을 알 수 있습니다. 마찬가지로 다음 3 자리 r-- 는 그 그룹에서 읽기만 가능하고 다음 3 자리 역시 다른 사람은 읽기만 가능하게 권한이 주어져 있음을 알 수 있습니다.

(참고로 그룹이란 stud 를 말합니다. 아까 pwd 명령으로 경로를 한번 보신적이 있으시죠.. home1/stud/lkach 즉 home1 디렉토리 밑의 stud 그룹의 lkach 계정이라고 할 수 있겠습니다. 그룹은 실제적으로 그다지 사용하지는 중요하지 않은 관계로 그냥 다른 사람 하고 비슷하거라고 생각하시고 자세한 사항은 책을 참조해 주세요 ^^ ;;)

이번에는 권한 설정을 변경해 보겠습니다. 각각의 3 자리는 고유한 값이 있습니다.

r w x <- 권한입니다.

4 2 1 <- 십진수 숫자입니다.

이 숫자들을 더해서 자신이나 그룹이나 다른사람들의 권한을 각각 숫자로 나타내시면 됩니다. 이를 어떻게 쓰는 것인지 예를 들어 보이겠습니다.

명령은 chmod 명령을 쓰시고 다음에 주고자 하는 권한을 숫자로 적으시면 됩니다. 예를들어 rwx 권한을 다주고 싶으시면 숫자 7입니다. 4+2+1 하시면 7 맞죠? 이걸 자신과 그룹과 다른사람에게 다 주게되면 777 입니다. 만약 자신은 모든 권한을 다갖고 그룹과 다른사람은 아무것도 못하게 하시려면 chmod 700 하시면 됩니다. 4+2+1 과 0+0+0 과 0+0+0 입니다. 만약 자신은 읽기 쓰기만 가능하고 그룹과 다른사람은 쓰기만 또 자신은 모든 권한을 다갖고 그룹과 다른사람은 실행과 읽기만 가능하게 하려면 chmod 755 하시면 됩니다. 4+2+1 과 4+0+1 과 4+0+1 해서 755 가 나온 것입니다. (참고로 700 보다는 755 정도로 해놓으셔야 홈페이지를 운영하실 경우 다른 사람이 홈페이지를 볼 수 있습니다.)

실제 예입니다.

```
$ ls -l
total 8
-rw-r--r--  1 lkach  stud    124  5월 13일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud    575  5월 13일  22:20 local.login
-rw-r--r--  1 lkach  stud    560  5월 13일  22:20 local.profile
-rw-r--r--  1 lkach  stud      0  5월 31일  11:42 testfile <- 이 파일
보이시죠
drwxr-xr-x  3 lkach  stud    512  5월 27일  21:34 www_home

$ chmod 777 testfile <- chmod 명령으로 testfile 이란 파일에 권한을
<-무조건 다 줬습니다.

$ ls -l
total 8
-rw-r--r--  1 lkach  stud    124  5월 13일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud    575  5월 13일  22:20 local.login
-rw-r--r--  1 lkach  stud    560  5월 13일  22:20 local.profile
-rwxrwxrwx  1 lkach  stud      0  5월 31일  11:42 testfile <-이렇게
바뀌었습니다.
drwxr-xr-x  3 lkach  stud    512  5월 27일  21:34 www_home
$
```

(10) 암호 바꾸기 입니다.

```
$ passwd <- passwd 라는 명령어를 쓰시면 됩니다.
passwd: Changing password for lkach
Enter login password: <- 여기에 현재 암호를 넣으신후
```

```

New password:                <- 새로운 암호를 입력하시고
Re-enter new password:      <- 새로운 암호를 다시한번 입력하시면
됩니다.
passwd (SYSTEM): passwd successfully changed for lkach
$

```

(10) 홈페이지 제작시 참고사항.

```

$ ls -l
total 8
-rw-r--r--  1 lkach  stud    124  5월 13일  22:20 local.cshrc
-rw-r--r--  1 lkach  stud    575  5월 13일  22:20 local.login
-rw-r--r--  1 lkach  stud    560  5월 13일  22:20 local.profile
-rwxrwxrwx  1 lkach  stud      0  5월 31일  11:42 testfile
drwxr-xr-x  3 lkach  stud   512  5월 31일  18:20 www_home
$

```

여기에서 www_home 디렉토리가 보이 실겁니다. 이 디렉토리 안에 제작하신 html 파일이나 기타 홈페이지 관련 파일이나 디렉토리를 넣으시면 됩니다. 파일을 올리실때는 ftp 프로그램을 다운 받아서 ftp 프로그램을 이용해서 자신의 계정에 파일을 올리시는 것이 좋습니다. ftp 프로그램은 무료 프로그램도 많이 있습니다.

www_home 디렉토리 안에는 기본적으로 index.htm 파일이 들어있습니다.

```

$ cd www_home
$ ls -l
total 4
-rw-r--r--  1 lkach  stud      0  5월 27일  21:34 index.htm
drwxr-xr-x  3 lkach  stud   512  5월 13일  22:20 jsp
$

```

그런데 이 inde.htm 은 그냥 기본적으로 들어가 있는 파일입니다. 파일 크기가 0 인 것으로 아실수 있습니다. 직접 제작하신 홈페이지의 처음 화면은 index.htm 이라는 이름으로넣어 주시면 됩니다. 그후에 홈페이지가 제대로 들어갔는지 확인 하시려면 http://iam.snut.ac.kr/~자신의 아이디 이렇게 직접 주소를 넣어서 확인해 보시면 됩니다. 제 아이디의 경우를 예로 들면 http://iam.snut.ac.kr/~lkach 가 되겠네요.