2019. 3. 23.

1차 과제 – Linux 기초

금요일 - 이성원 교수님

2015722087 컴퓨터정보공학부

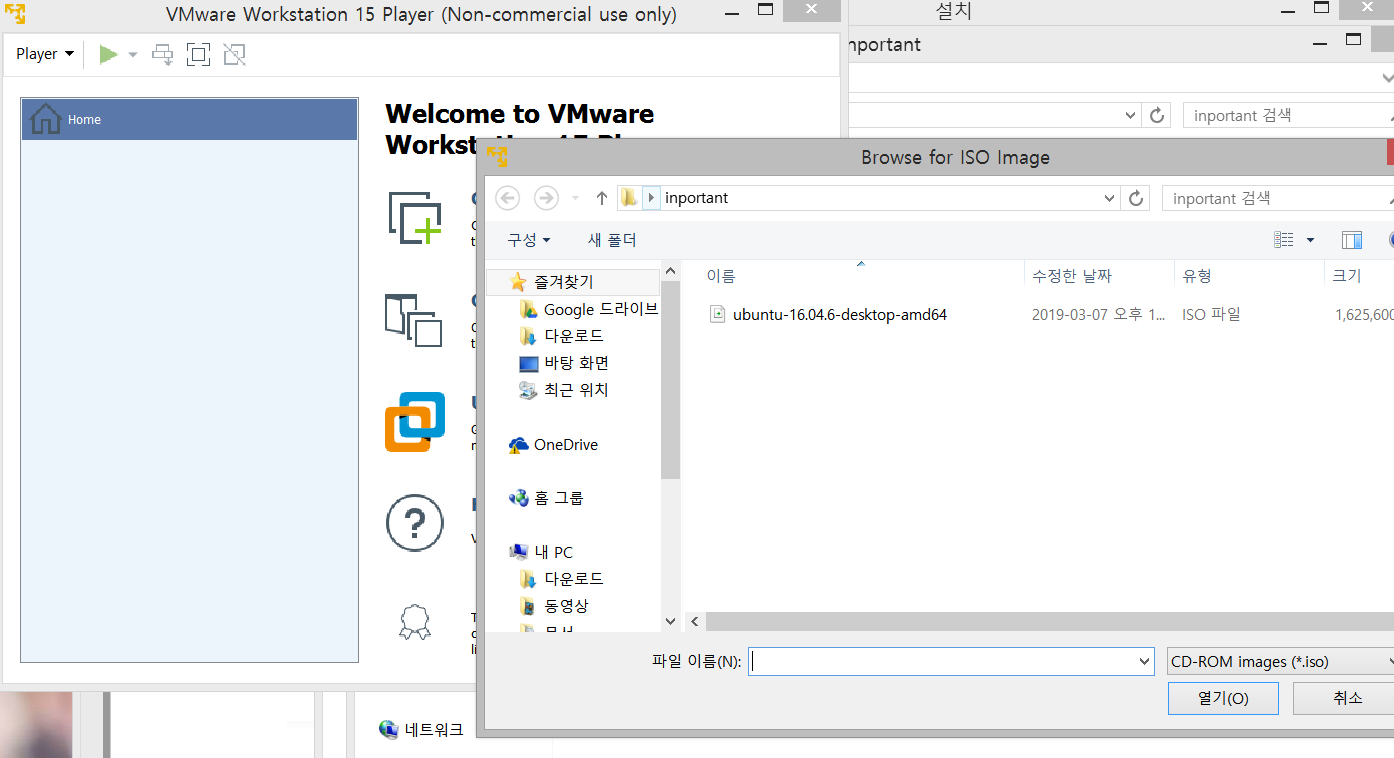
김민철

* 1. Ubuntu Installation

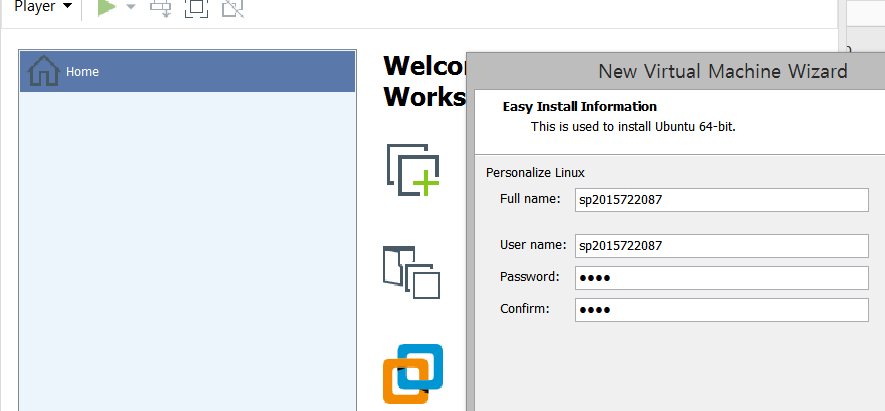
**Introduction**

* 이번 과제는 VMware 라는 단일 가상 머신 실행기를 이용하여, Linux 운영 체제를 Windows PC 내에 추가적으로 설치하여 Windows 운영체제가 설치된 PC 에서 Linux 운영체제를 실행할 수 있는 환경을 만드는 것이다.

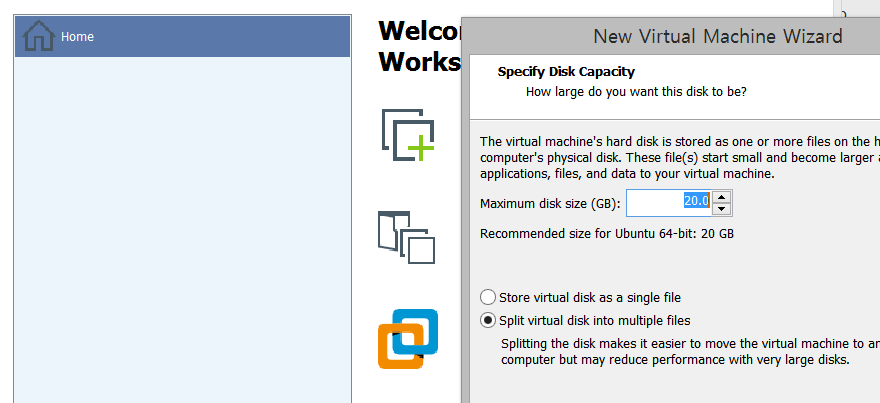
**Result**



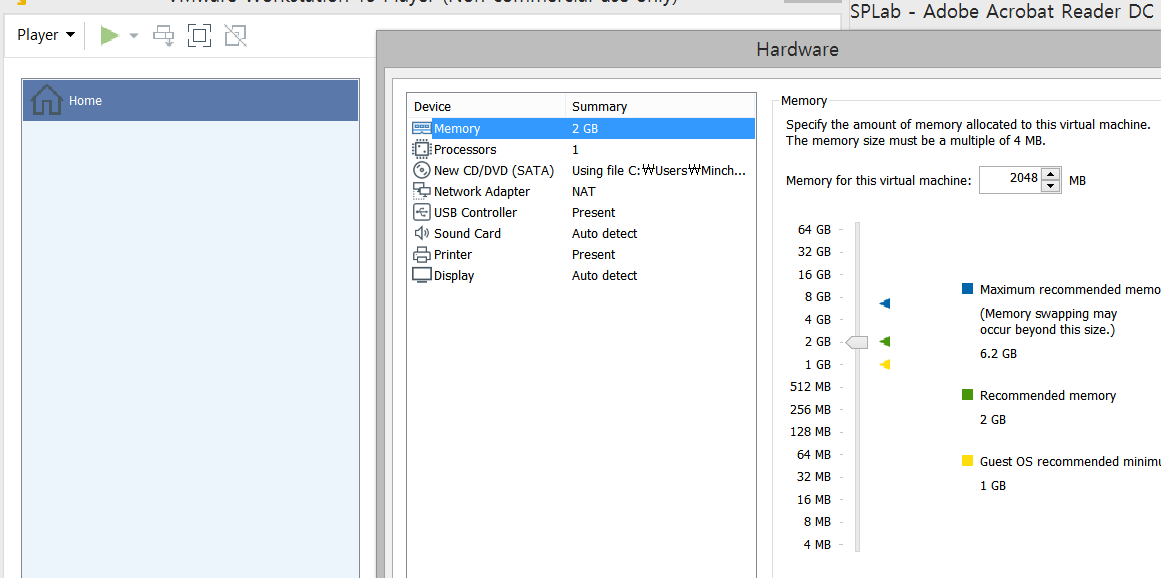
* VMware 에서 Linux의 배포판 중 하나인 우분투를 선택하여 설치를 시작한다.



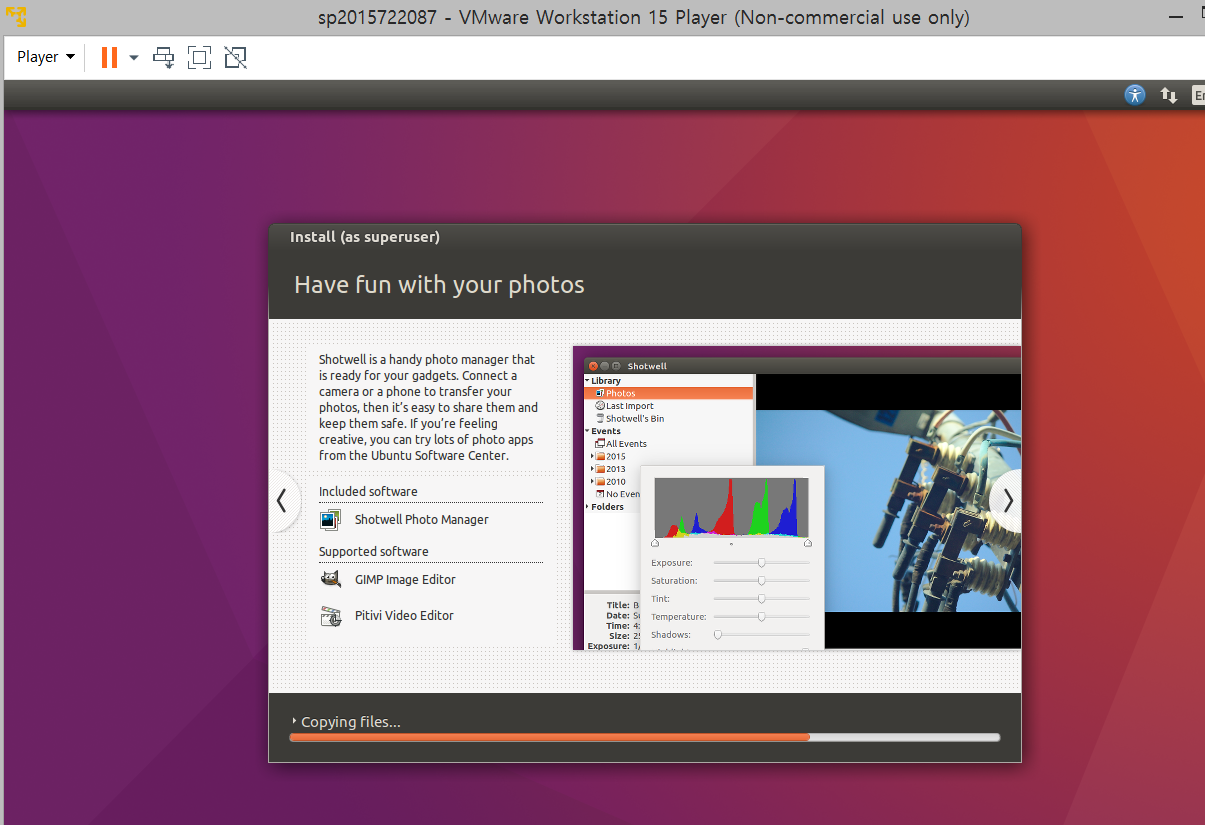
* 우분투에서 로그인 할 계정과 패스워드를 입력한다.



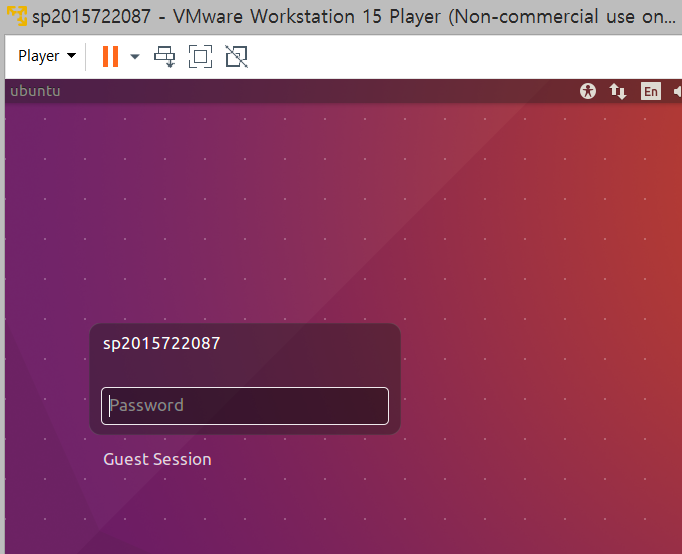
* Windows PC에서 동작하는 가상머신에 할당할 저장장치(하드디스크) 용량을 설정한다.



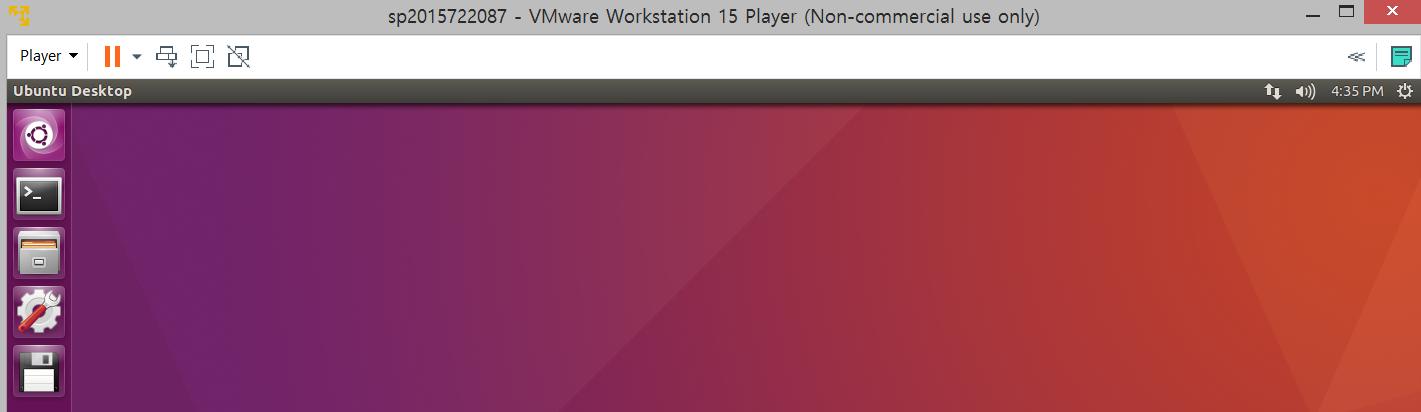
* Windows PC에서 동작하는 가상머신에 할당할 저장장치(RAM)의 용량 및 CPU의 코어 수 등 하드웨어 설정을 선택한다.



* Windows PC 에서 동작하는 가상 머신을 이용하여 우분투를 설치하는 과정이다.



* 설치가 끝나고 User ID 와 Password를 입력하는 창이다.

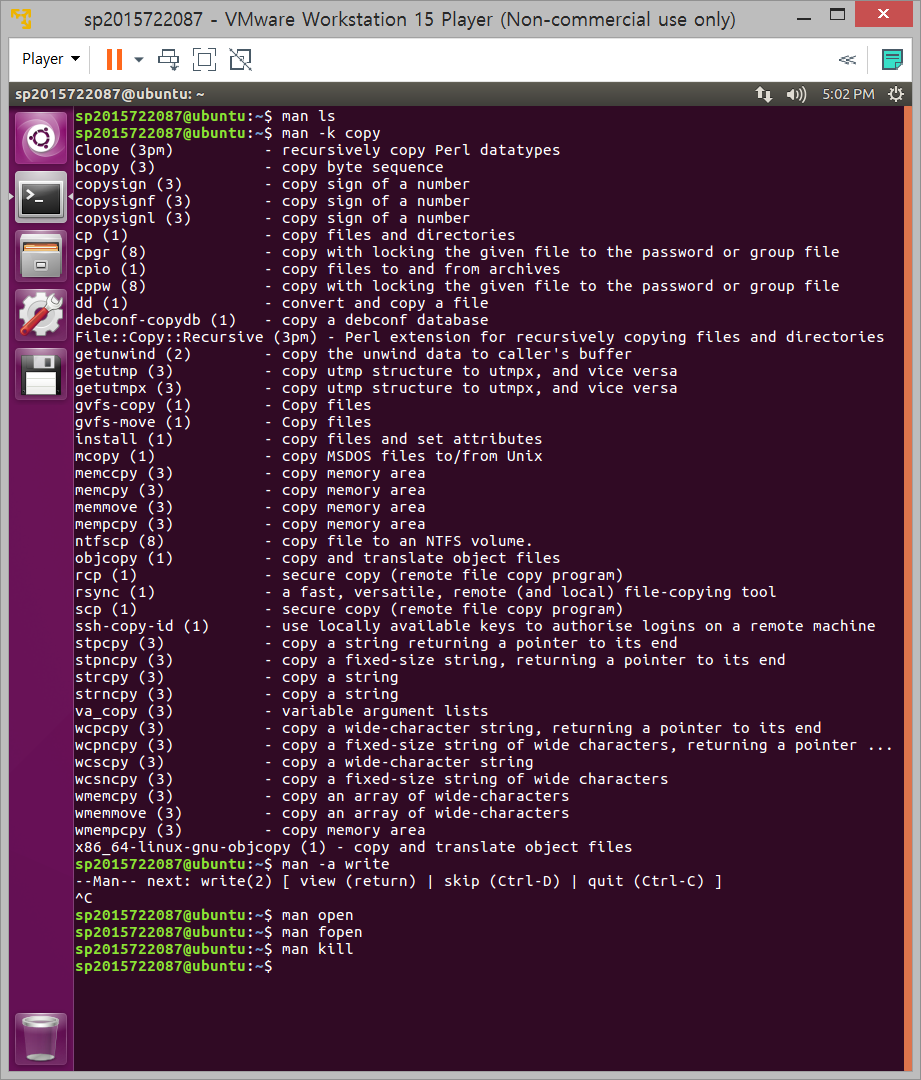


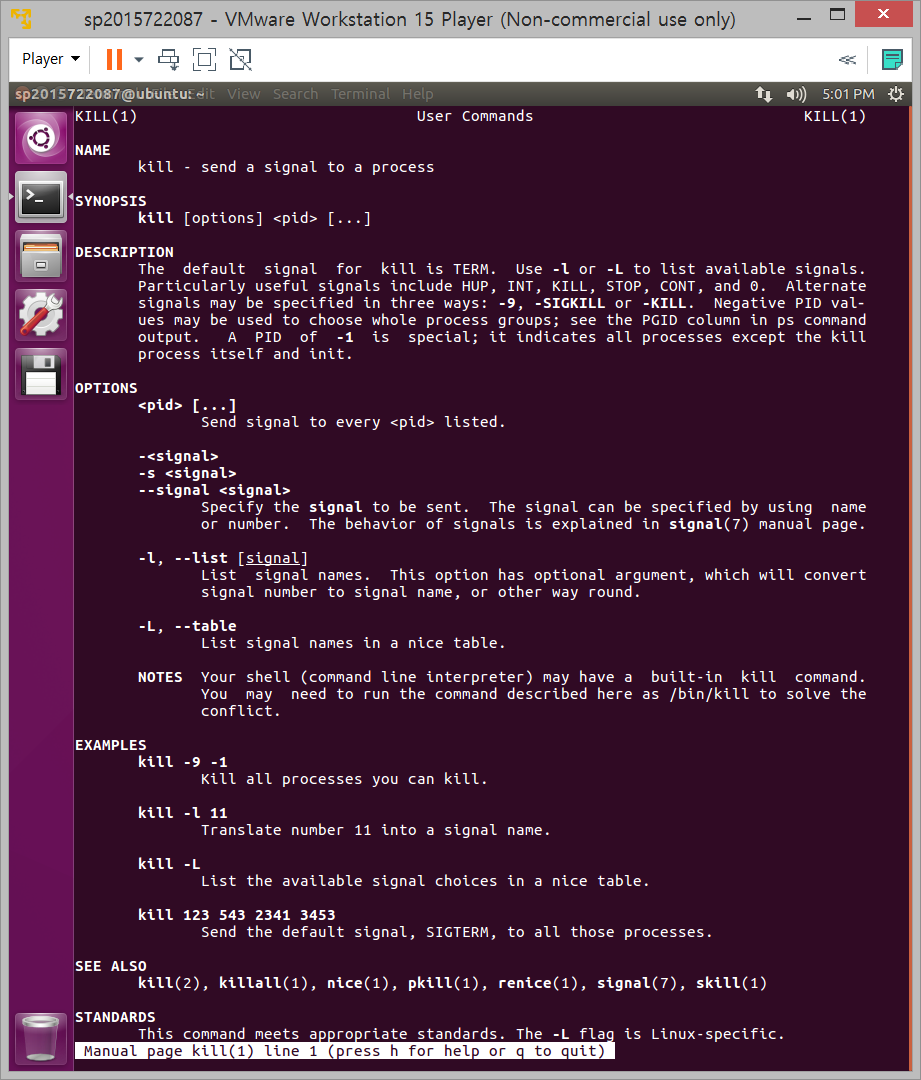
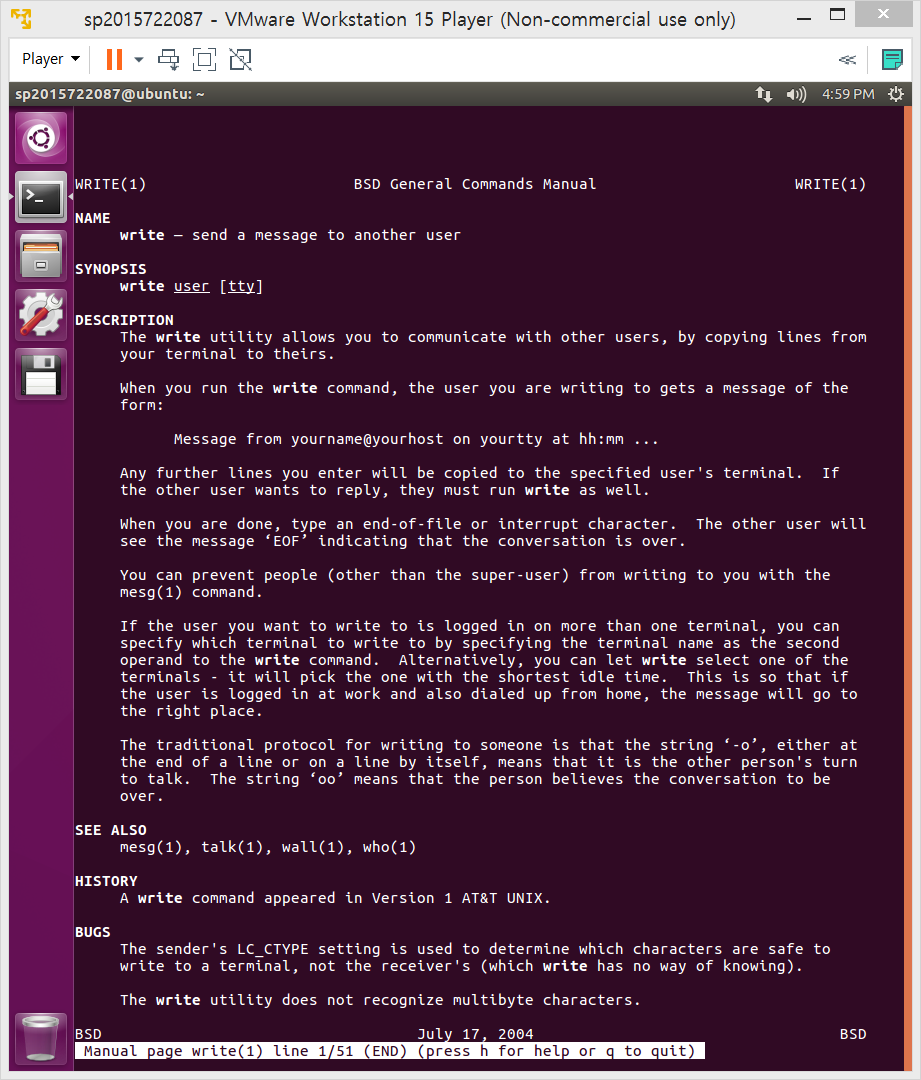
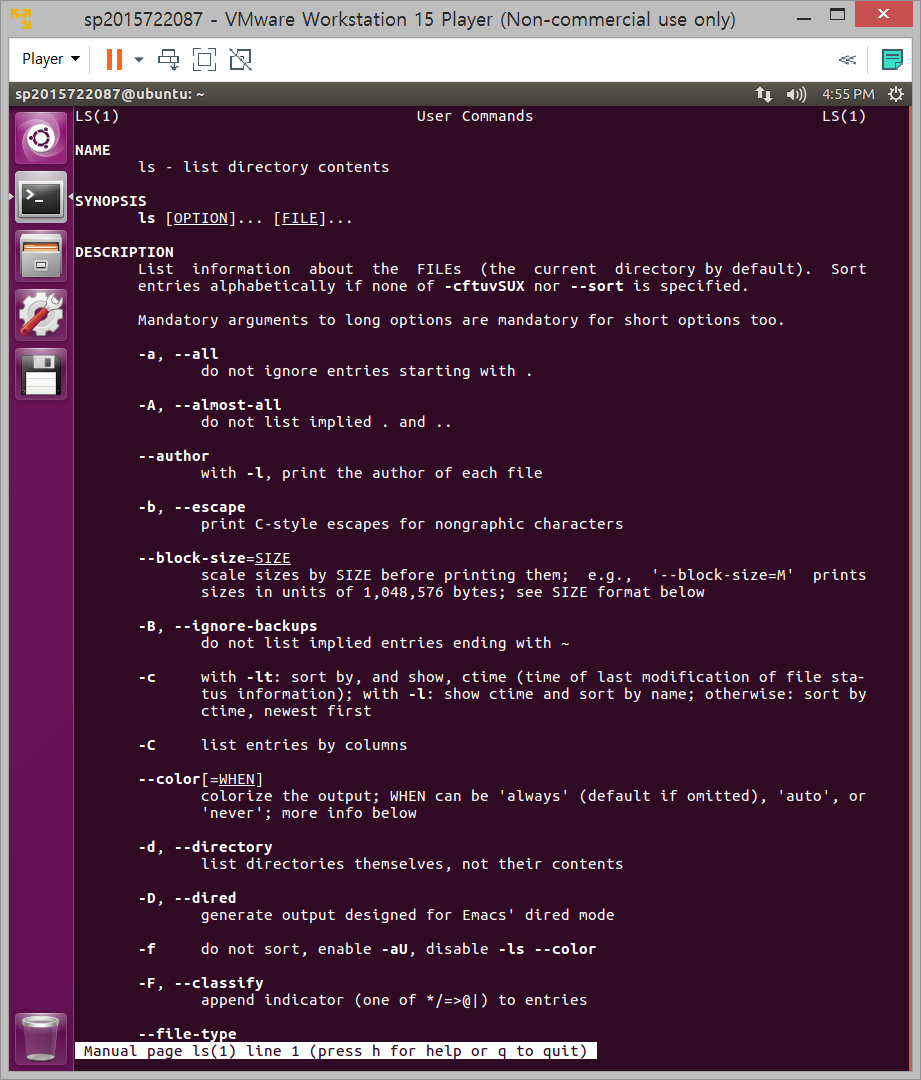
* 설치가 완료되어 우분투가 실행된 모습이다.
  1. Usage of Linux Commands

**Introduction**

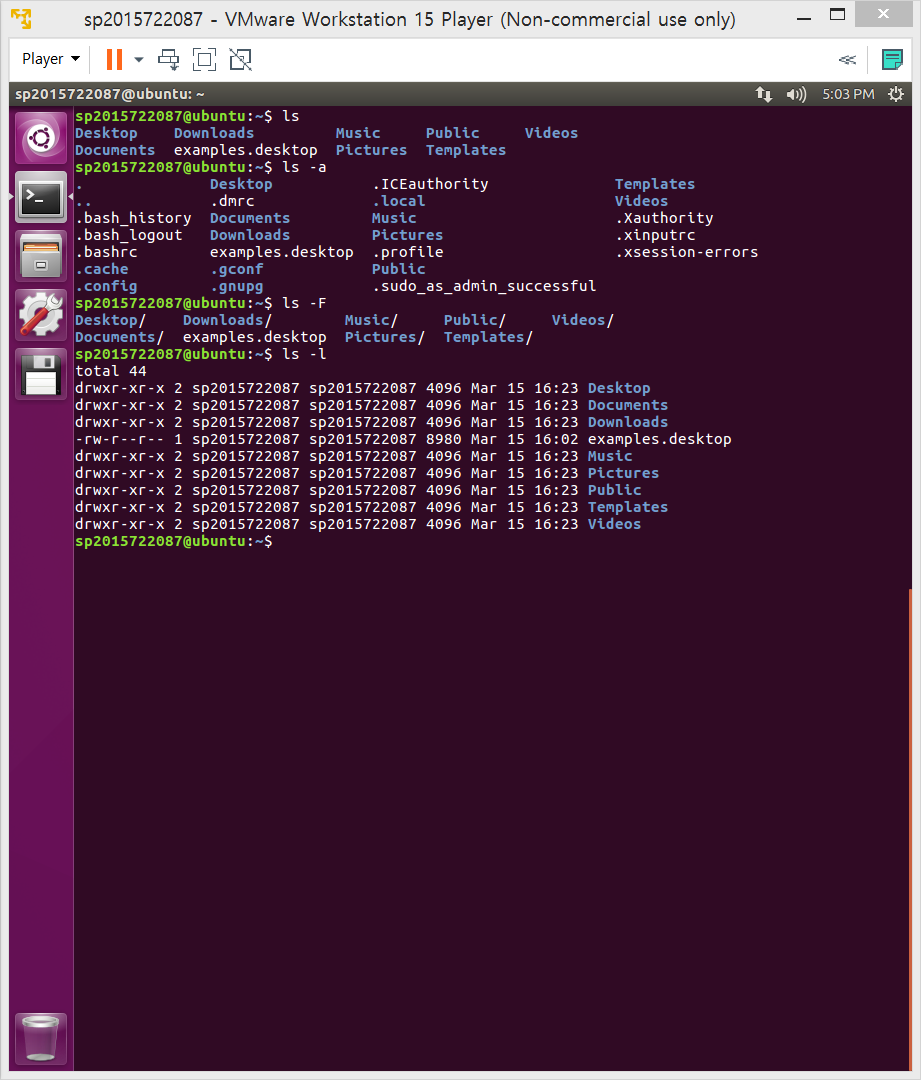
* 이번 과제는 실습 시간에 배운 Linux 명령어를 이해하고, 직접 명령어를 사용하여 각 명령어의 사용법과 각 명령어의 특징, 활용법 등을 습득하여 앞으로 Linux를 이용하여 실습을 진행함에 있어 명령어를 적절하게 사용할 수 있게 하는 것이다.

**Result**

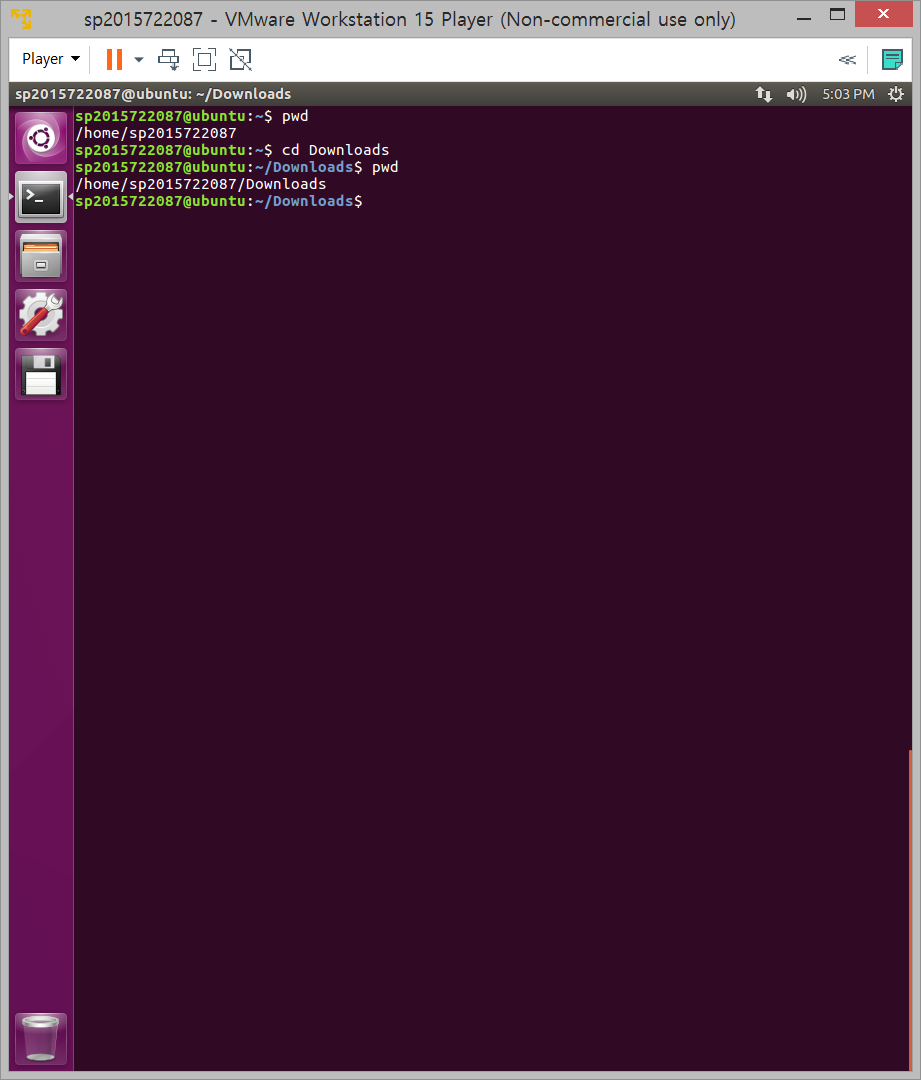




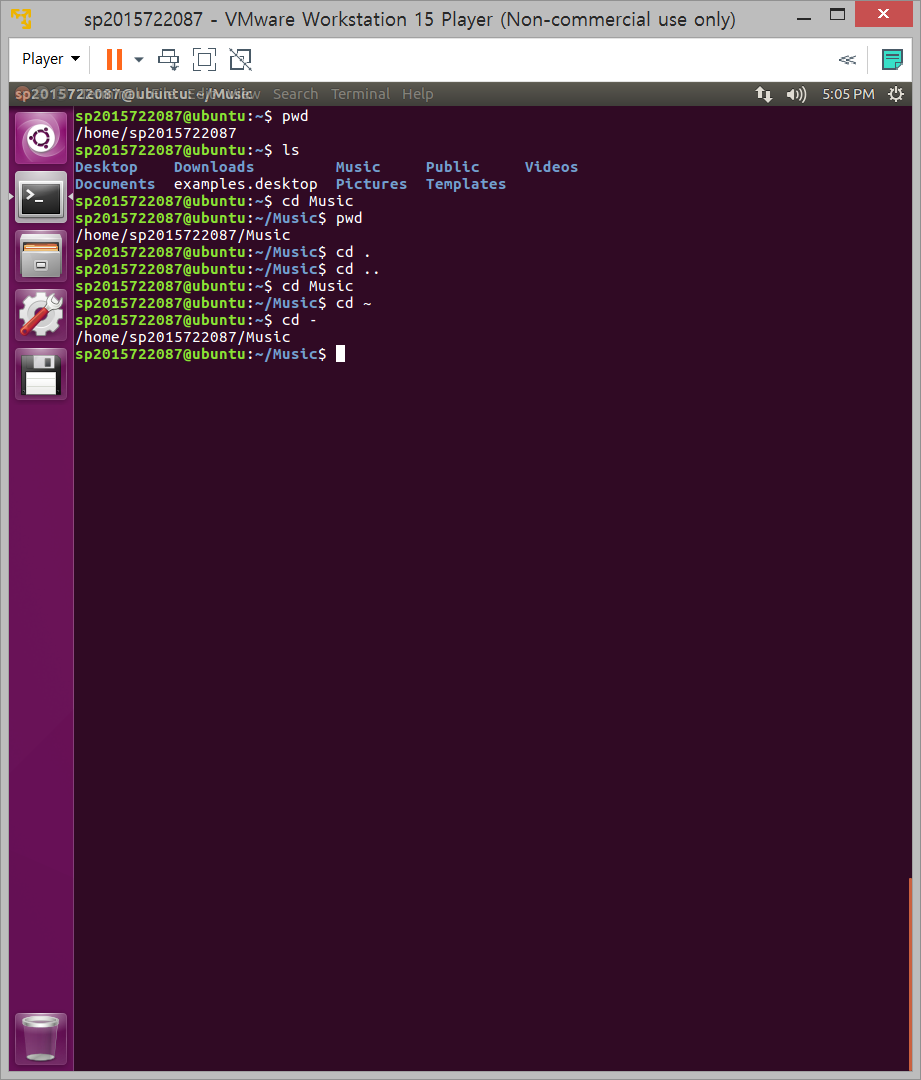
$ Man – man 이란 manual page 의 줄임말이다. 즉, 화면에 리눅스 명령어에 대한 매뉴얼 페이지를 출력해주는 명령어 이다. 명령어는 man [option] name 이렇게 입력하며, 옵션 란에 –k 를 입력하면 keword search를 진행하여 결과를 출력하라는 명령이다. 즉 $man –k copy 는 ‘copy’를 키워드 검색한 결과를 보여준다. 옵션에 –a 를 입력하면 검색한 매뉴얼 페이지 전체를 출력하라는 명령이다. $man –a write 는 write 함수에 관한 모든 매뉴얼을 출력한다.



$ ls – 현재 작업중인 디렉토리의 파일이나 폴더를 화면에 출력하라는 명령어이다. 옵션으로 –a를 입력할 경우 숨겨진 파일을 포함한 모든 파일을 보여주며, -F를 입력할 경우 파일의 종류를 구분하여 폴더에는 / 를, 실행파일에는 \* 를 표시해준다. –I를 입력할 경우에는 파일의 정보를 자세히 출력하는데, 이는 윈도우에서의 파일 보기 메뉴 중 자세히 보기와 비슷하게 파일의 상세한 속성을 모두 보여준다.



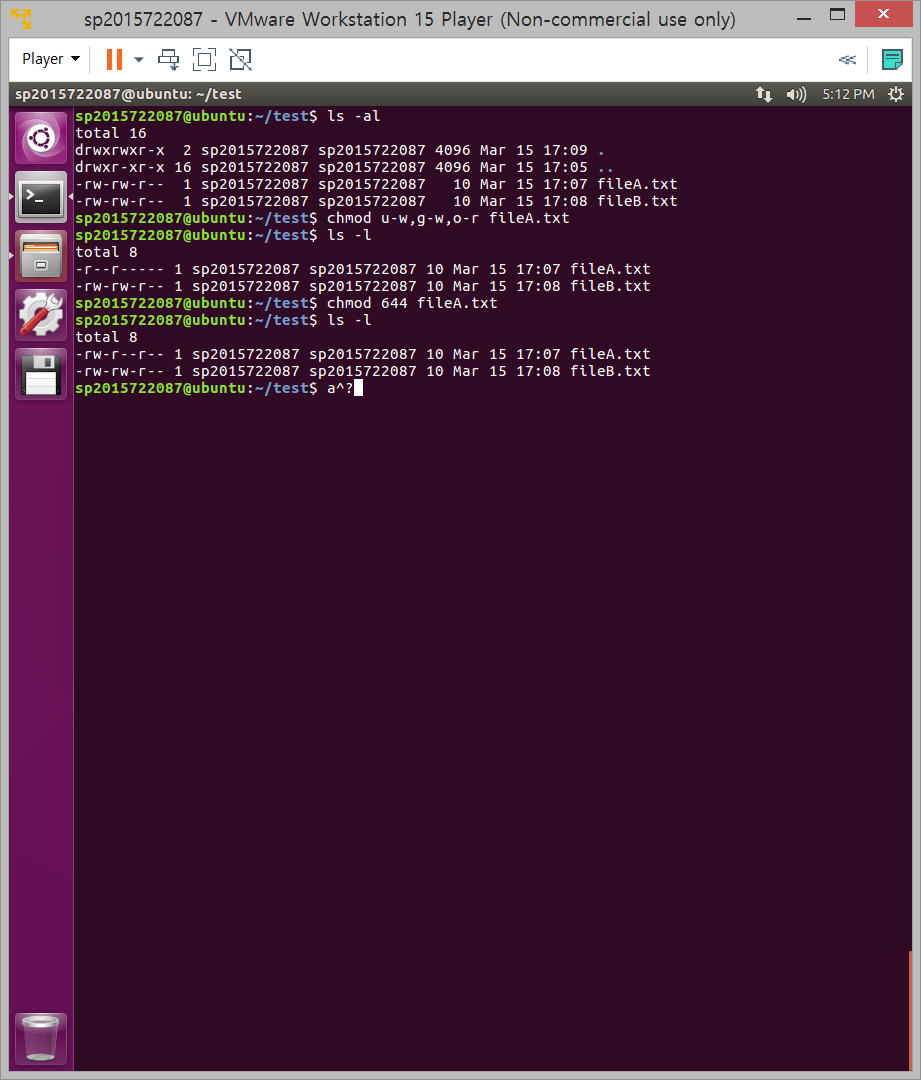
$ pwd – print working directory 이다. 현재 작업중인 디렉토리의 절대 경로를 출력한다.



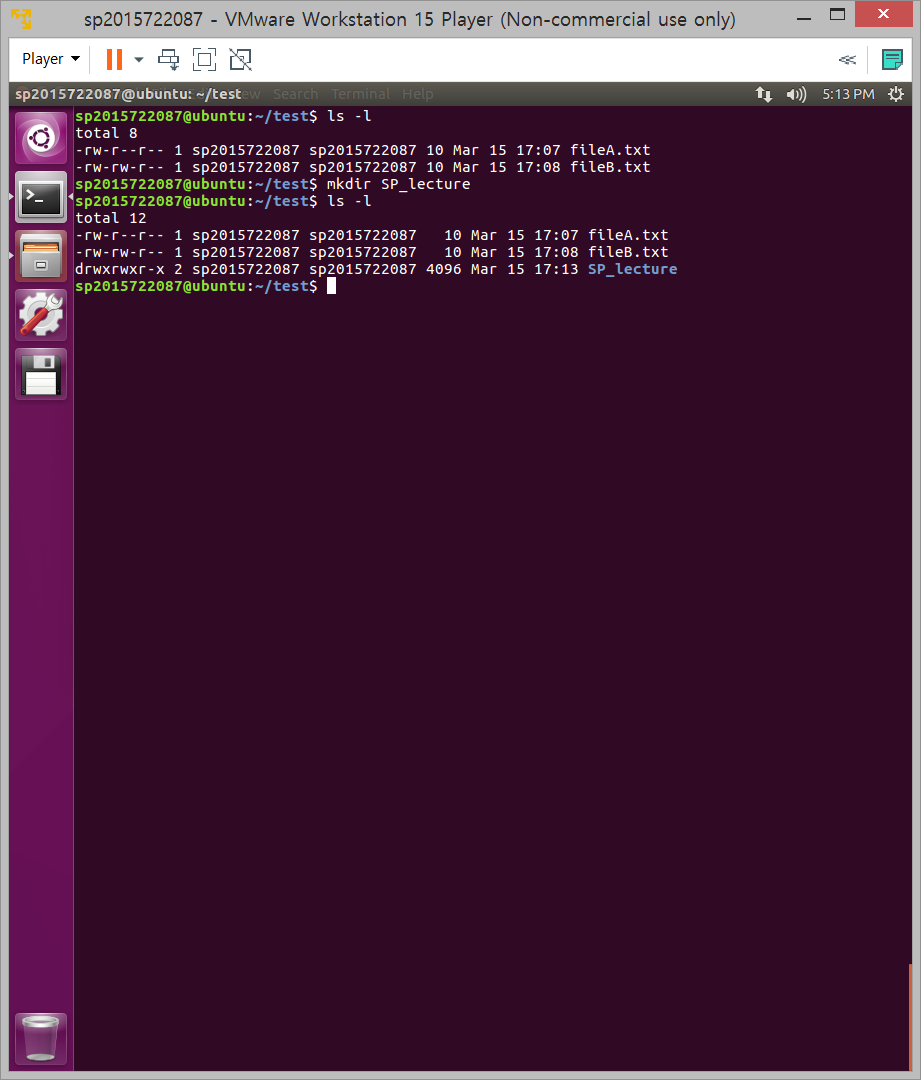
$ cd – change directory 이다. cd 뒤에 현재 디렉토리의 내에 있는 폴더를 입력하면 그 폴더의 위치로 이동하고, cd 뒤에 ‘.’ 은 현재 디렉토리, ‘..’ 은 상위 디렉토리, ‘/’ 는 최상위 디렉토리, ‘~’ 는 사용자 홈 디렉토리, ‘-’ 는 직전에 작업하던 디렉토리로 이동한다.



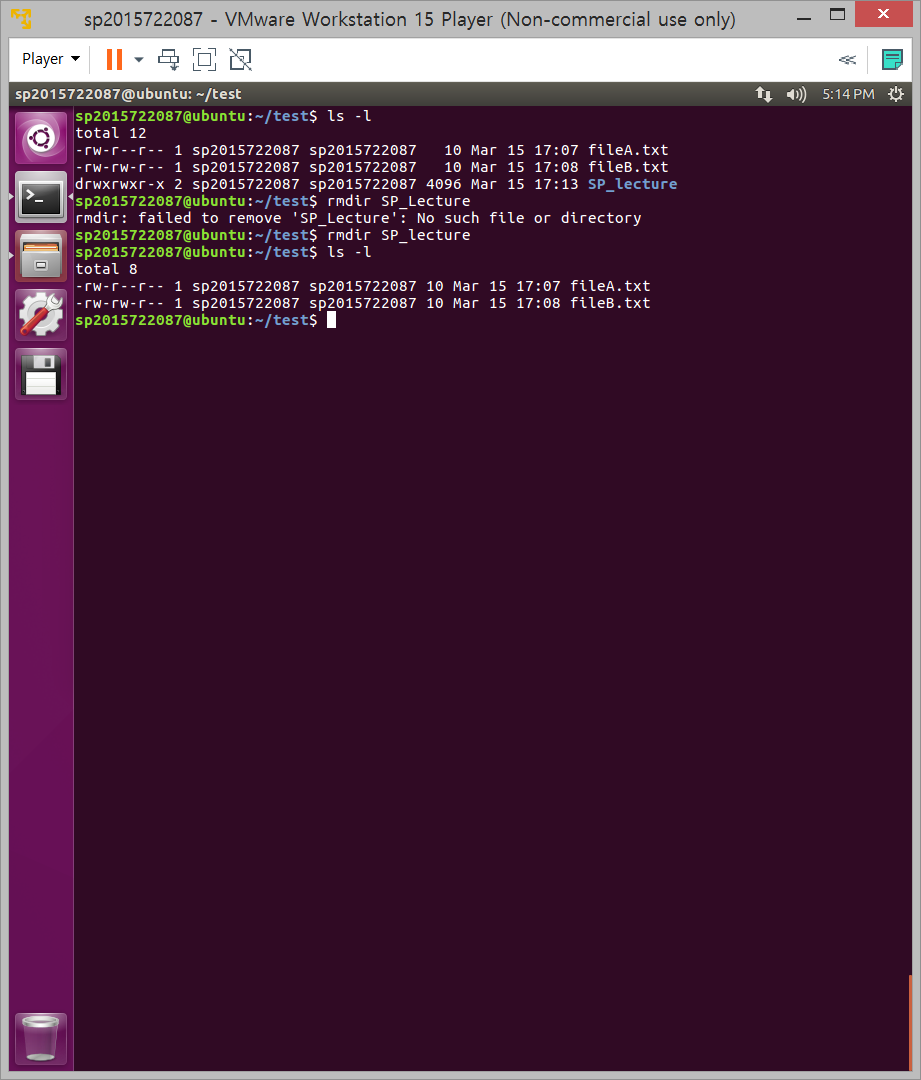
$ cat – 파일을 연결하거나 파일의 내용을 화면에 출력하는 명령어이다. 옵션 입력없이 Cat 명령어 뒤에 파일명을 입력할 경우 해당 파일의 내용을 출력하고, 파일명을 두개 입력할 경우 두 파일의 내용을 연결하여 화면에 출력한다.



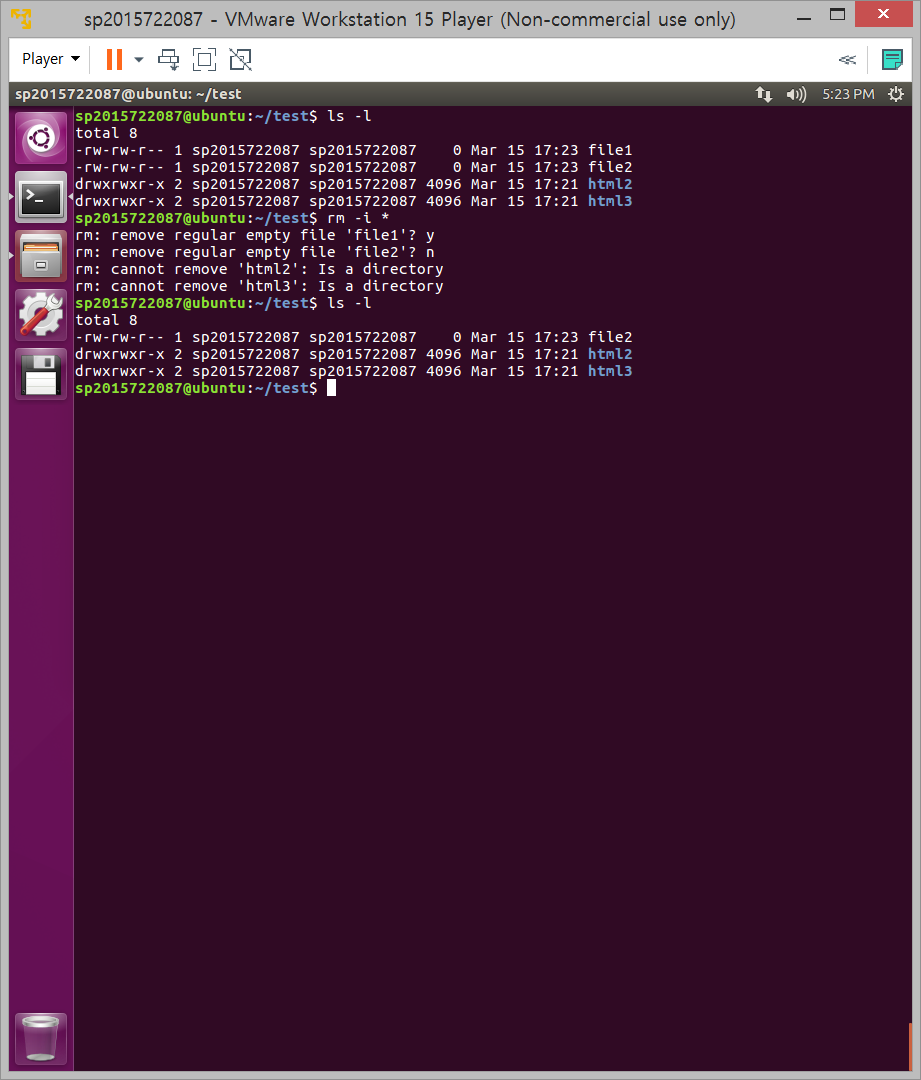
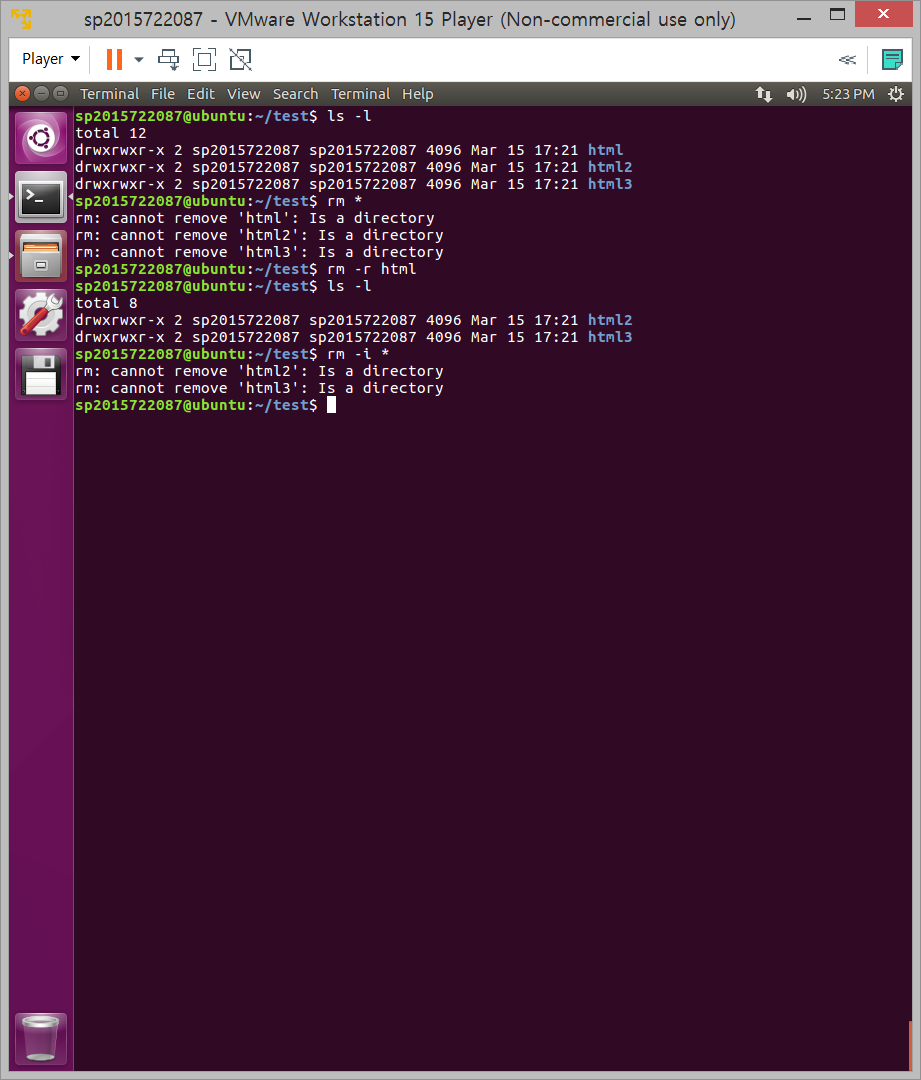
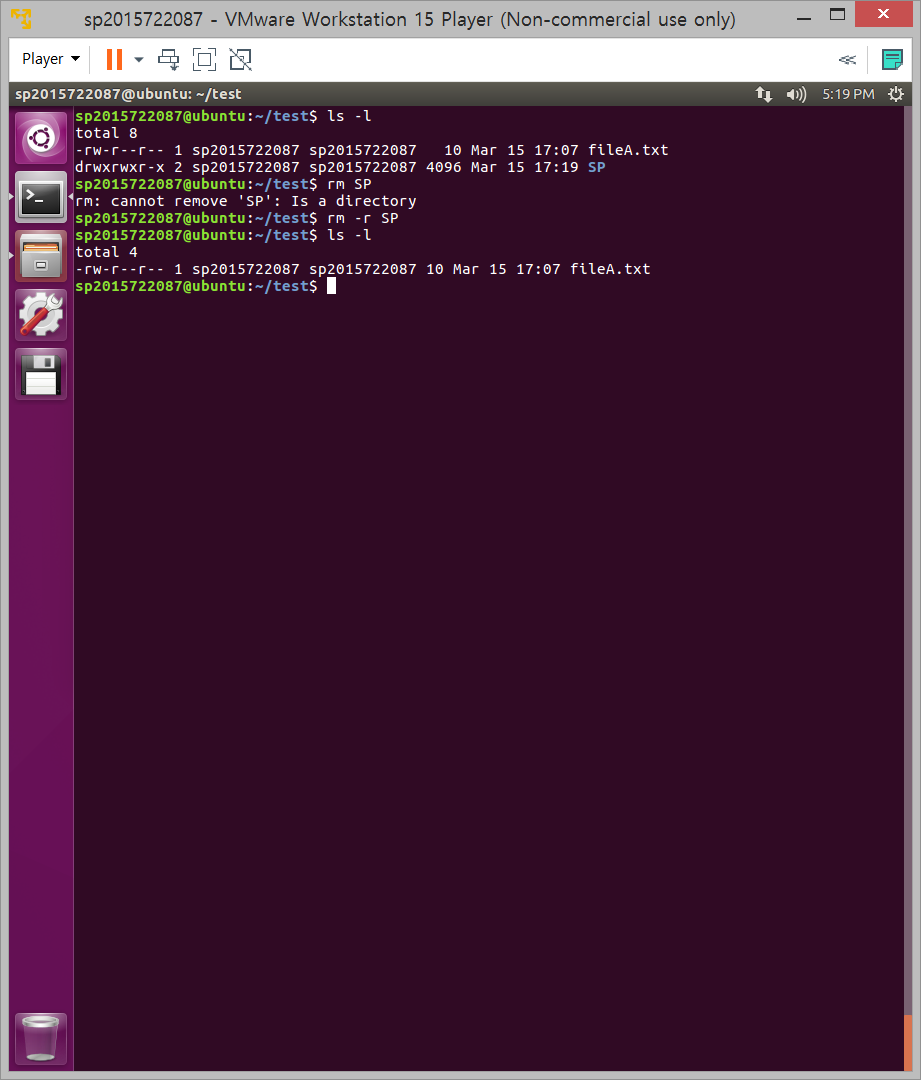
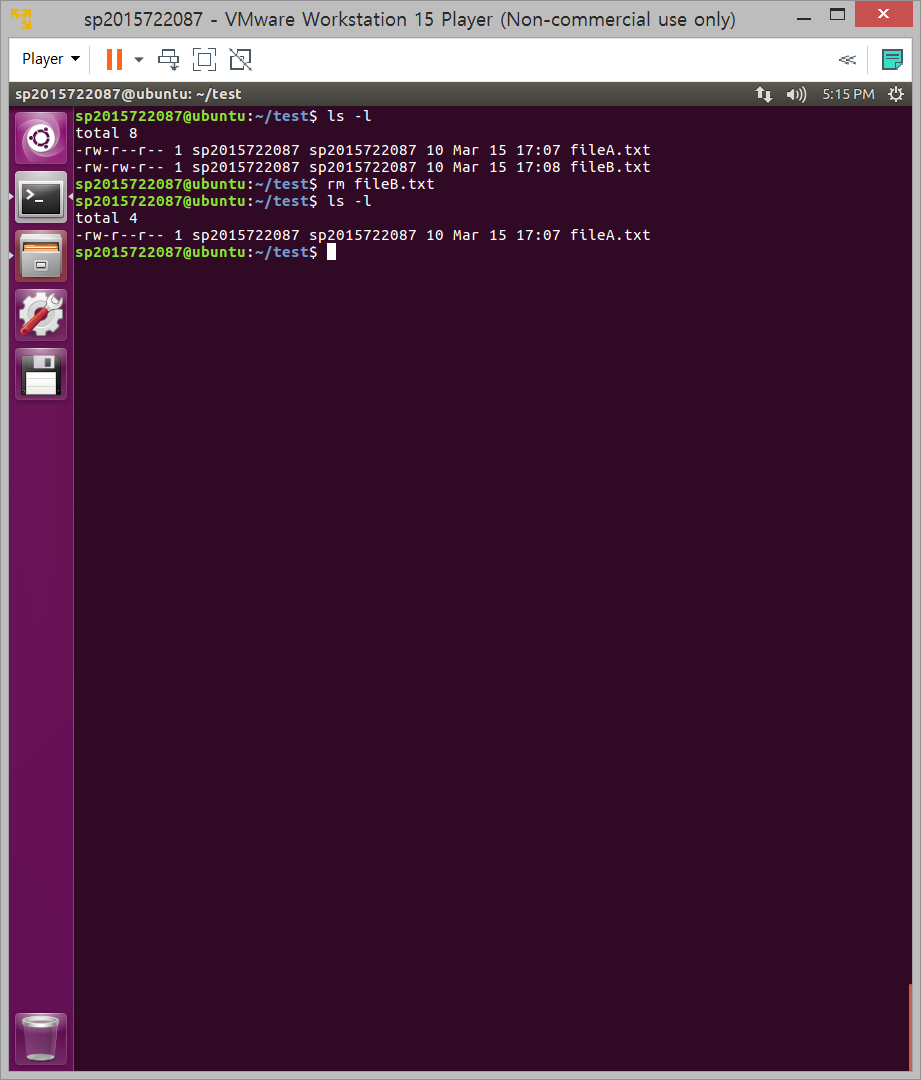
$ chmod – 파일의 권한을 변경하는 명령어이다. User – Group – Other 순서로 나오며, 읽기, 쓰기, 실행의 권한을 조정할 수 있다. 3개비트로 표현하기위해 100(4) – 읽기, 010(2) – 쓰기, 001(1) – 실행 권한을 의미하고 모든 권한을 주는 것은 111(7) 이다. 모든 사용자에게 모든 권한을 주려면 chmod 777 (파일명) 을 해주면 된다. 또한 위에서 입력한 것 처럼 숫자가 아닌 u-w나 g-w 처럼 입력하여 권한을 삭제시킬 수도 있다.



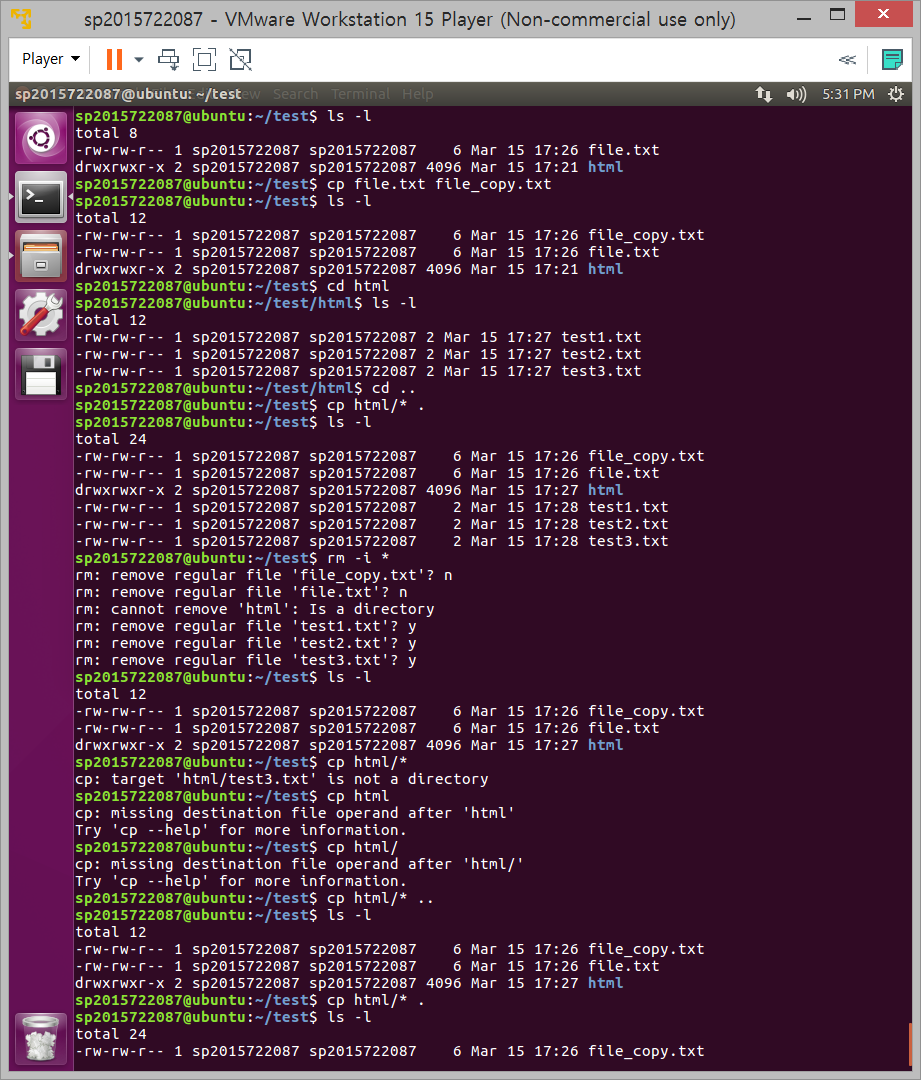
$ mkdir – make directory 의 줄임말이다. 새로운 빈 디렉토리를 만드는 명령어이며, 옵션을 입력하지 않을 경우, 지정한 이름으로 현재 디렉토리에 새로운 디렉토리를 만든다.



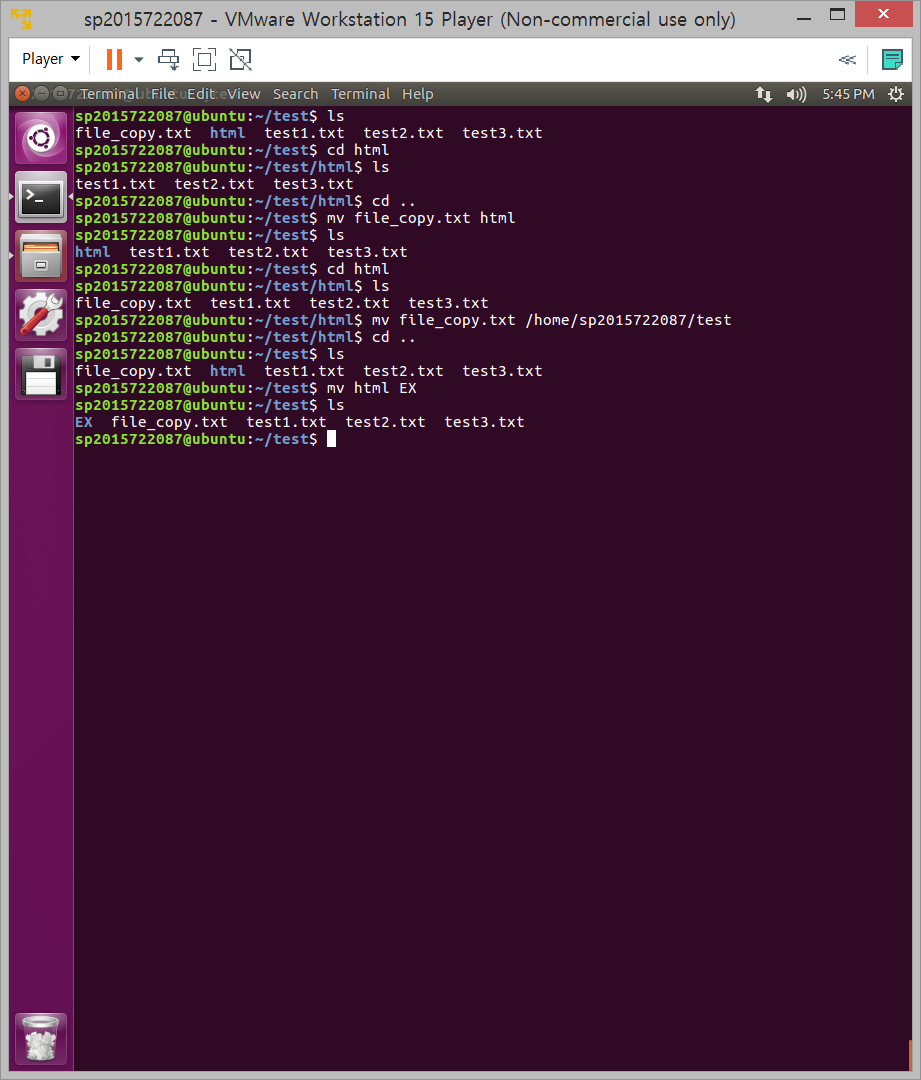
$ rmdir – remove directory의 줄임말이다. 빈 디렉토리를 지우는 명령어이며, 옵션없이 입력 할 경우 입력한 폴더의 디렉토리가 비어있지 않거나, 해당 폴더가 존재하지 않을 경우 오류가 발생한다. 이때 대문자와 소문자도 구분해야한다.



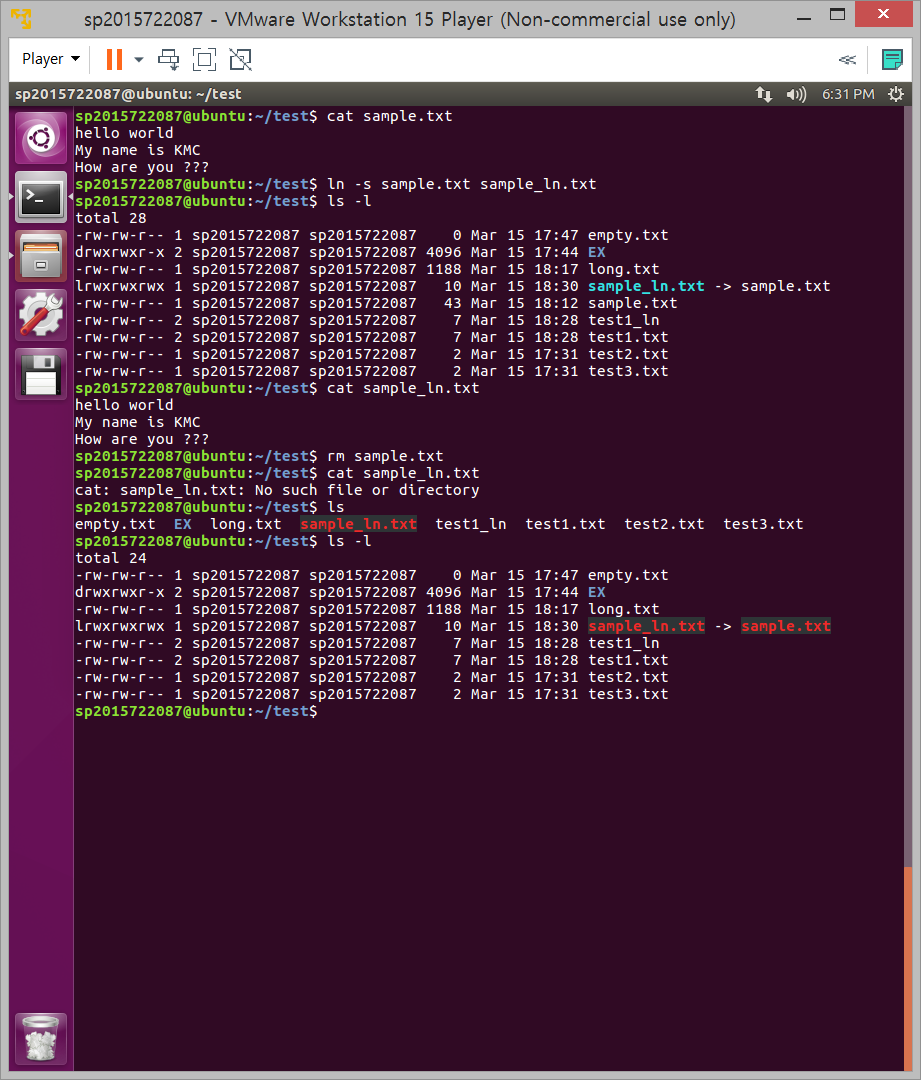
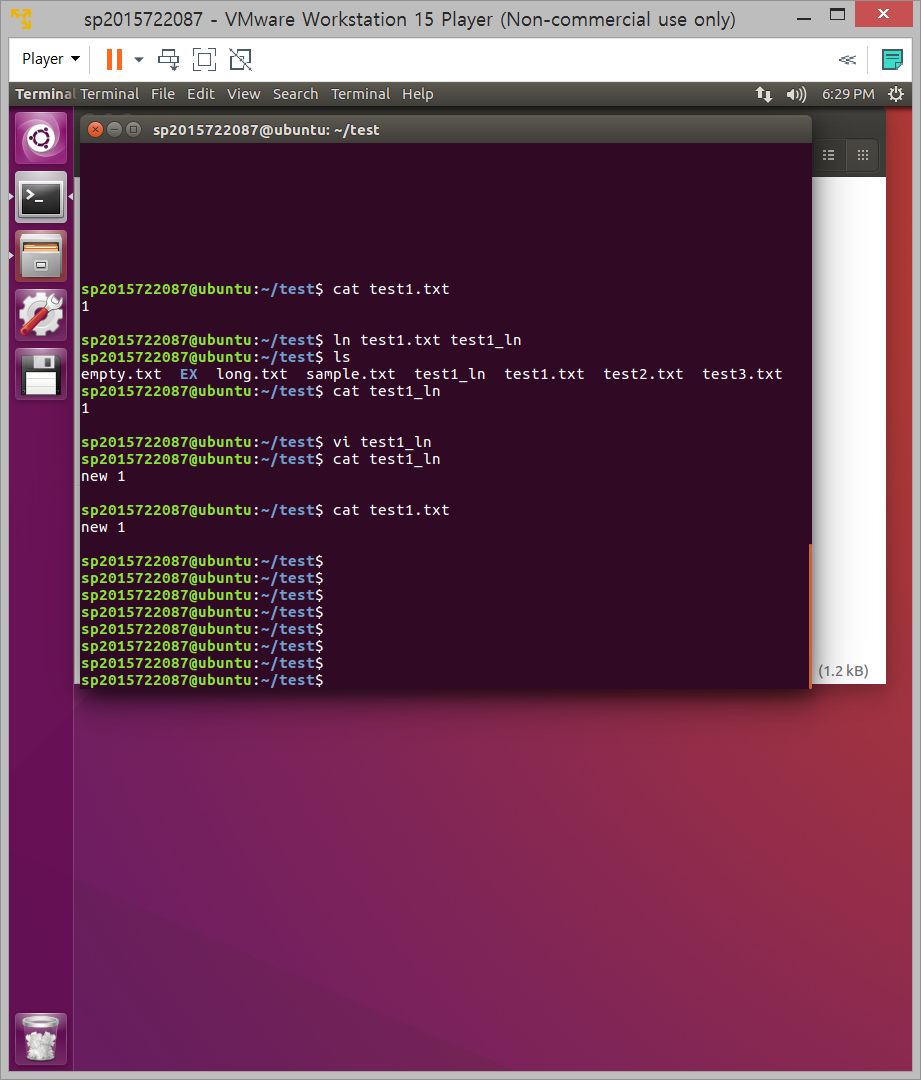
$ rm – remove의 줄임말이다. 옵션 없이 rm 뒤에 파일명을 입력할 경우 삭제한다. 파일명이 아닌 폴더명을 입력할 경우 삭제되지 않으며, 이때 옵션으로 –r 을 입력할 경우 폴더(해당 폴더안의 모든 파일 포함)도 삭제한다. 파일명이 아닌 ‘\*’을 입력할 경우 현재 디렉토리의 모든 파일(폴더 제외)을 삭제한다. 옵션으로 -i 를 입력한 경우에는 삭제하기 전에 삭제할 것인지 사용자에게 묻고 삭제한다.



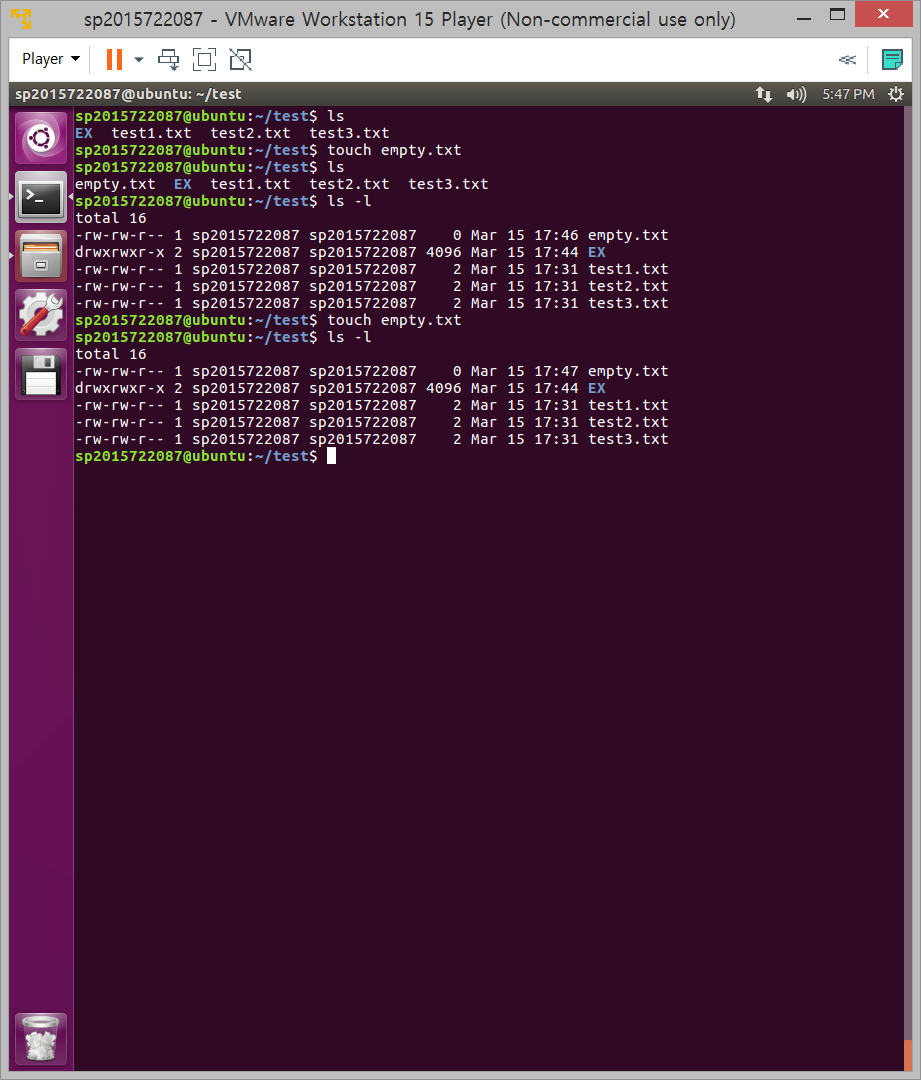
$ cp – copy의 줄임말이다. Cp 뒤에 복사할 파일명과 복사된 파일명을 입력하여 복사하며, 폴더명을 입력하고 /\*(해당 폴더내의 모든 파일) .(현재 디렉토리) 를 입력하면 해당 폴더내의 모든 파일을 현재 디렉토리에 복사한다. 완전한 독립적인 파일의 복사이기 때문에 원본을 지우거나, 복사본을 지운다고해서 서로 변화가 생기지 않는다.



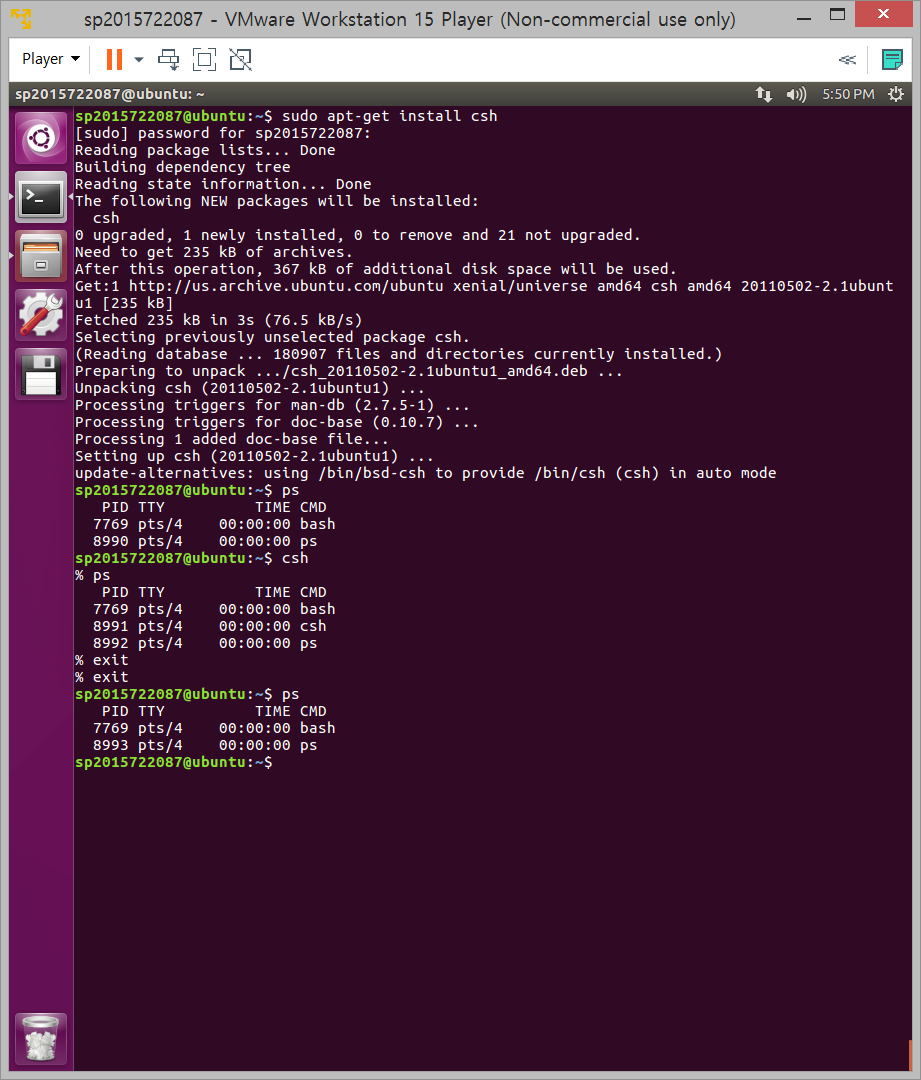
$ mv – 파일을 이동시키거나 파일(또는 폴더)의 이름을 변경시키는 명령어이다. 명령어 뒤에 파일명을 입력하고 그 뒤에 디렉토리를 입력하면 해당 디렉토리로 파일을 이동시키며, 명령어 뒤에 폴더명을 입력하고 그 뒤에 다른 폴더명을 입력할 경우 폴더명이 변경된다.



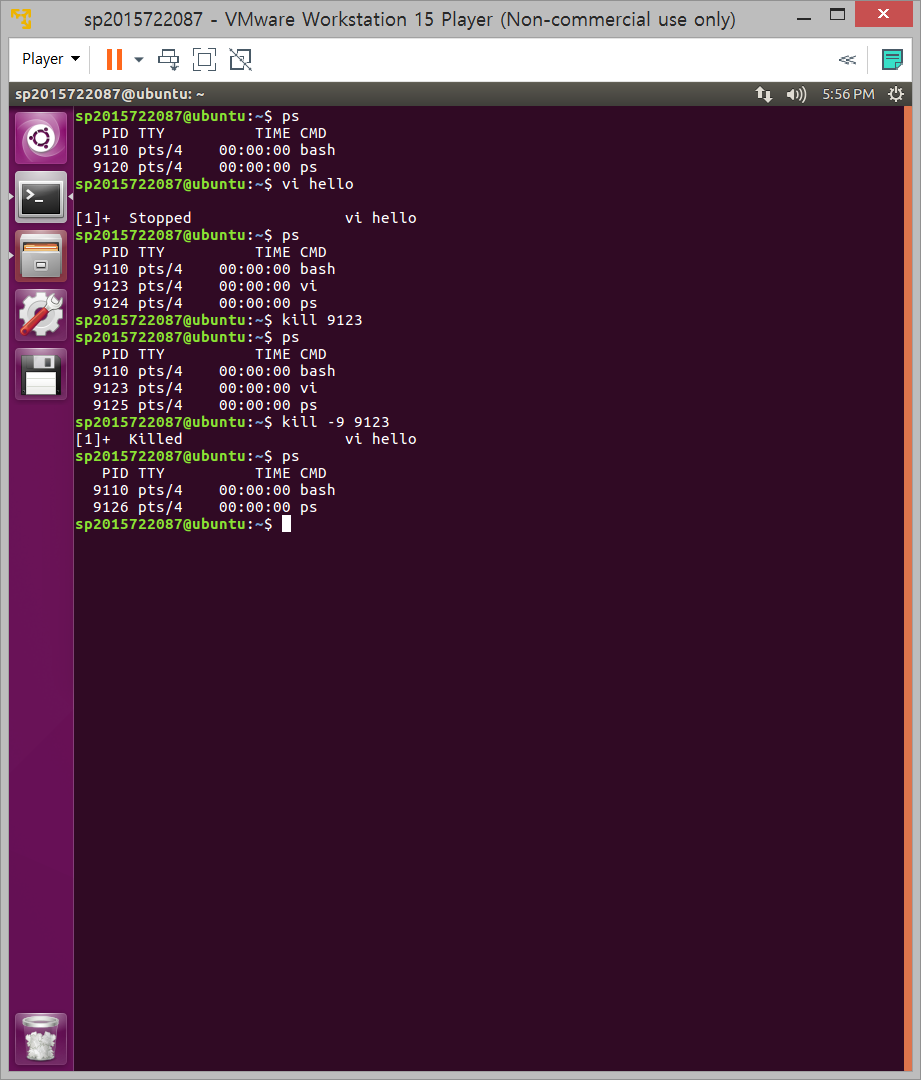
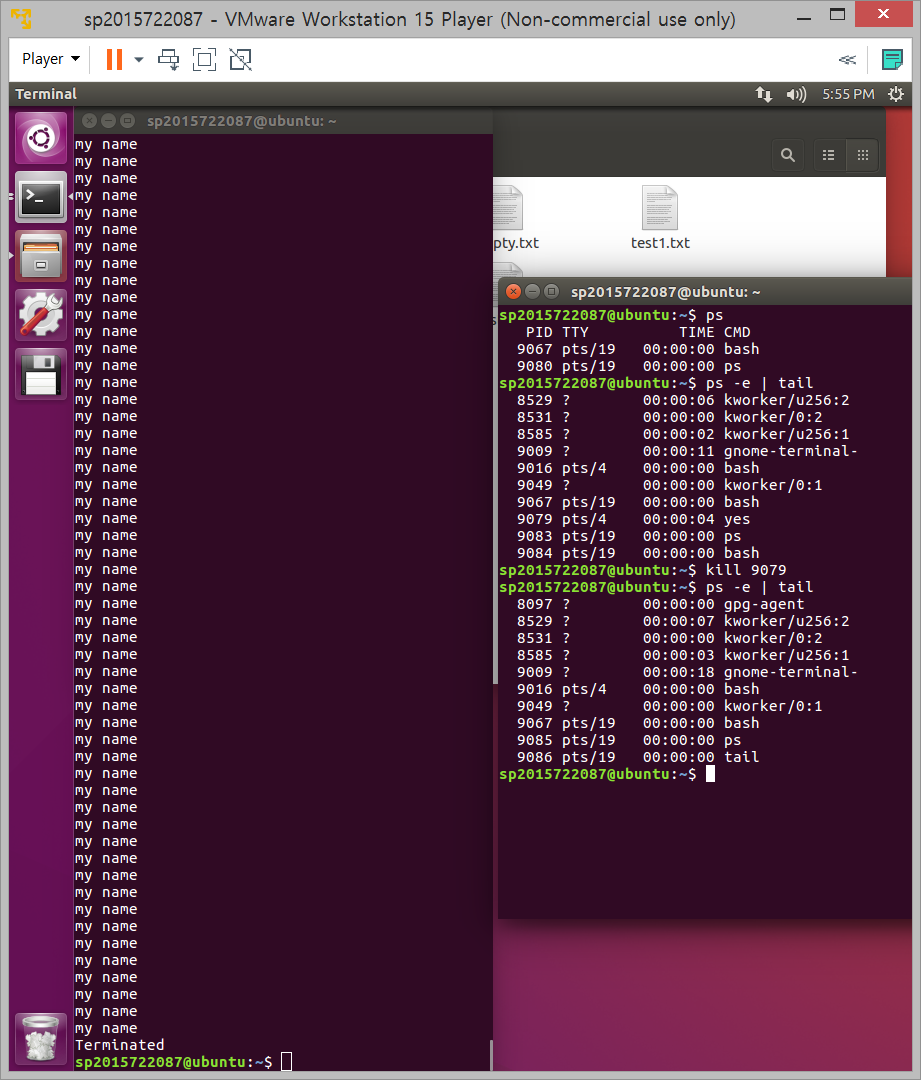
$ ln – hard link라고 하며, 파일들을 연결해주는 명령어인데, $ln 원본파일 링크할파일명 을 입력하면 원본파일과 같은 inode를 가리키는 링크할 파일명으로 파일이 생성되는데 이를 실행하면 원본과 똑같아서 뭐가 원본인지 구분이 힘들다. 파일 하나를 수정하면 나머지 파일도 똑같은 inode를 가리키기 때문에 모든 파일이 수정이 된다. Cp 명령어는 아에 같은 데이터로 독립된 파일을 생성하는 것이기 때문에 다르고, soft link인 symbolic link는 파일의 데이터를 가리키는 것이 아니라, 파일의 디렉토리를 데이터로 가지는 파일이기 때문에, 원본 파일이 존재하지 않아도 생성이 가능하다.



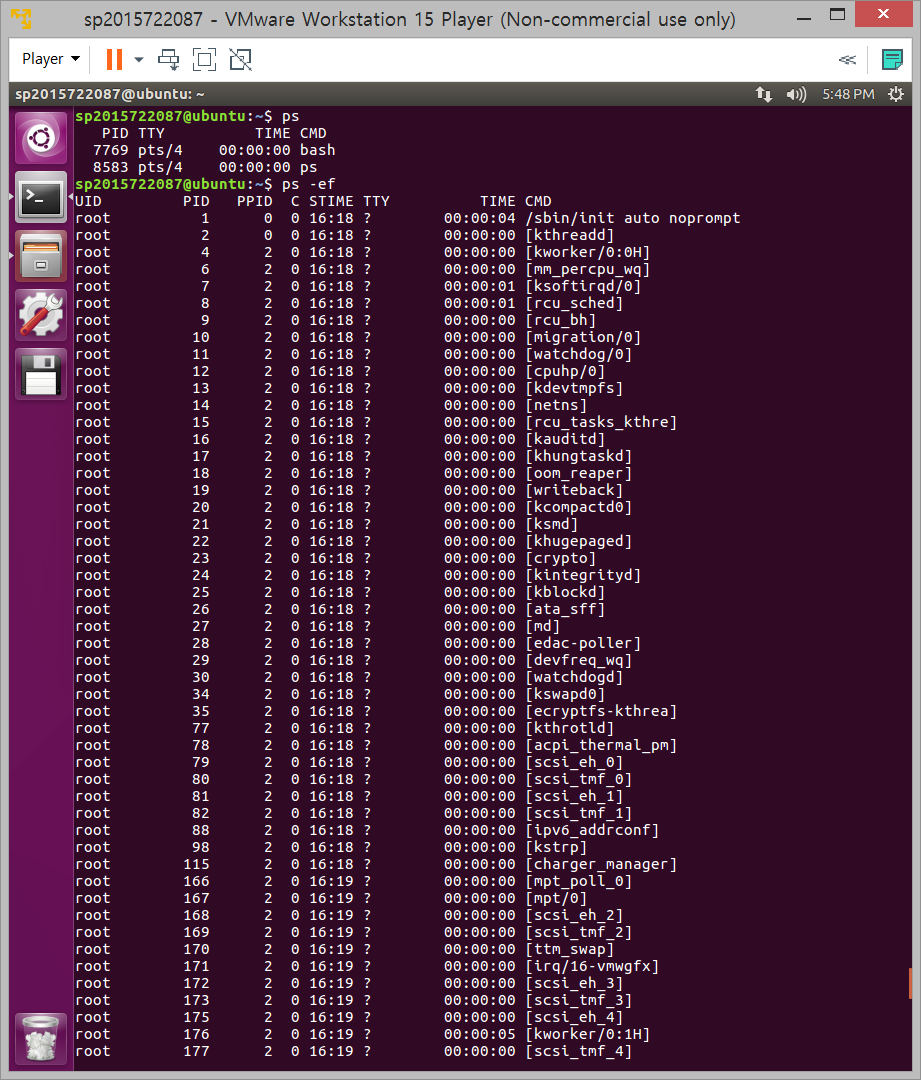
$ touch – 아무것도 작성되지 않은 파일을 생성하거나 최근 사용 시간을 변경하는 명령어이다. Touch 명렁어 뒤에 파일명을 입력하면 그 파일명으로 새 파일이 생성된다.



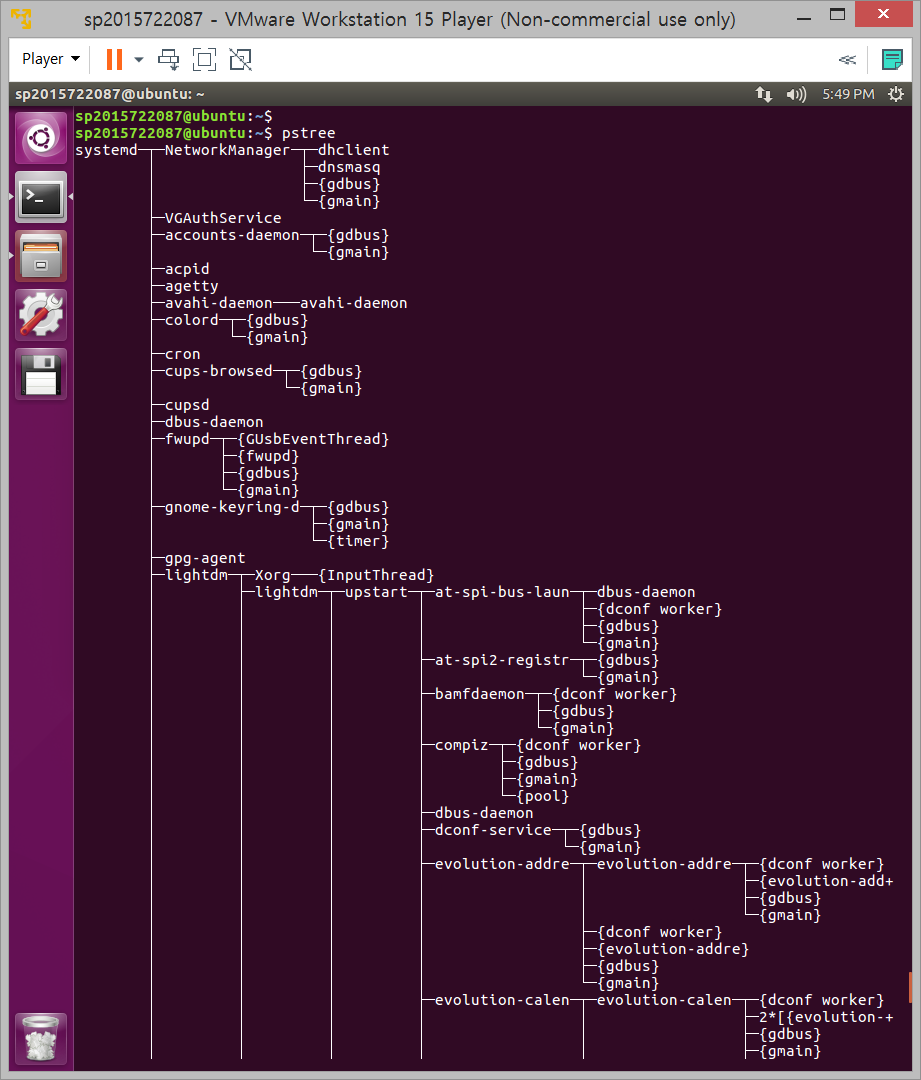
$ exit – 프로세스를 닫거나, shell (터미널)을 닫거나 할 때 사용하는 명령어이다.



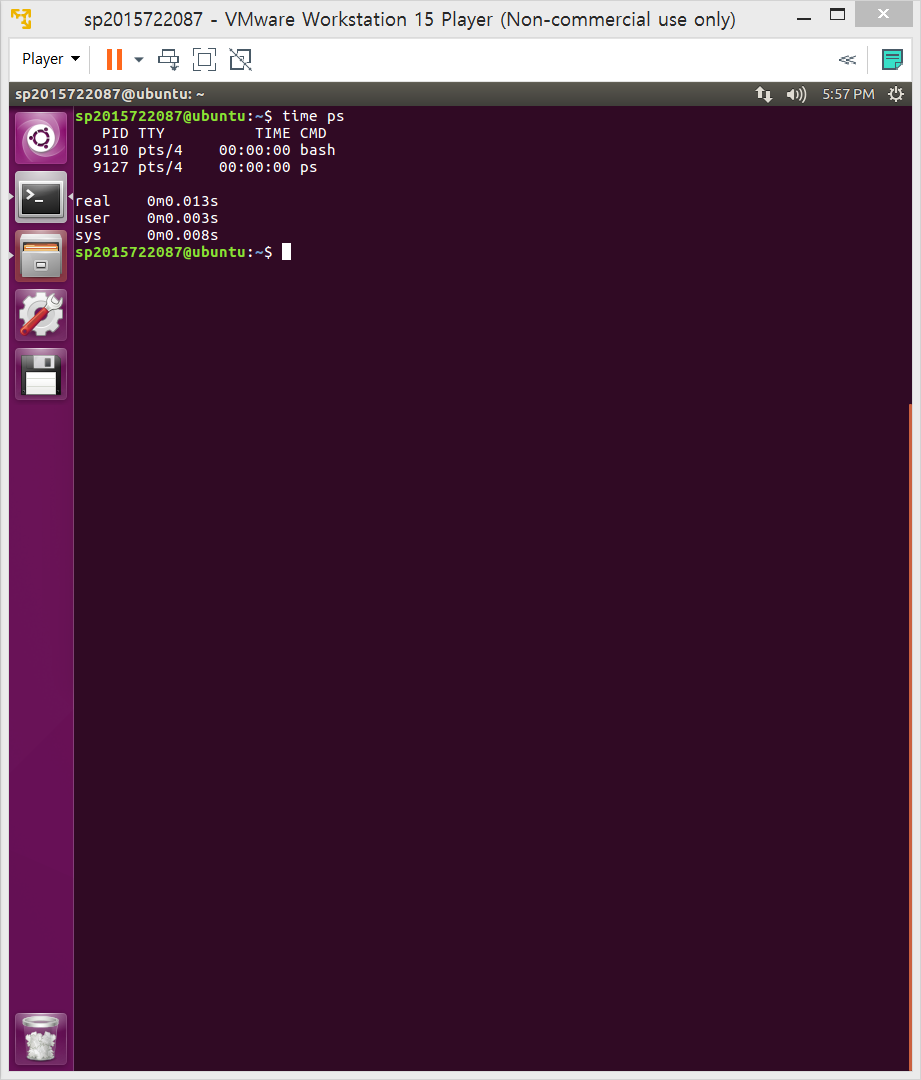
$ kill – 프로세스에 신호를 보내는 명령어이다. 신호를 받은 프로세스는 신호를 무시하거나, 미리 정해진 동작을 하거나, 개발자가 따로 지정한 동작을 한다. Default로 설정된 신호는 프로세스를 종료시키라는 것이다. Kill 명령어 뒤에 프로세스 이름을 입력하면 해당 프로세스에게 종료하라는 신호를 보내 프로세스가 종료된다. 옵션 값으로 -9를 입력하면 강제 종료 신호이다.



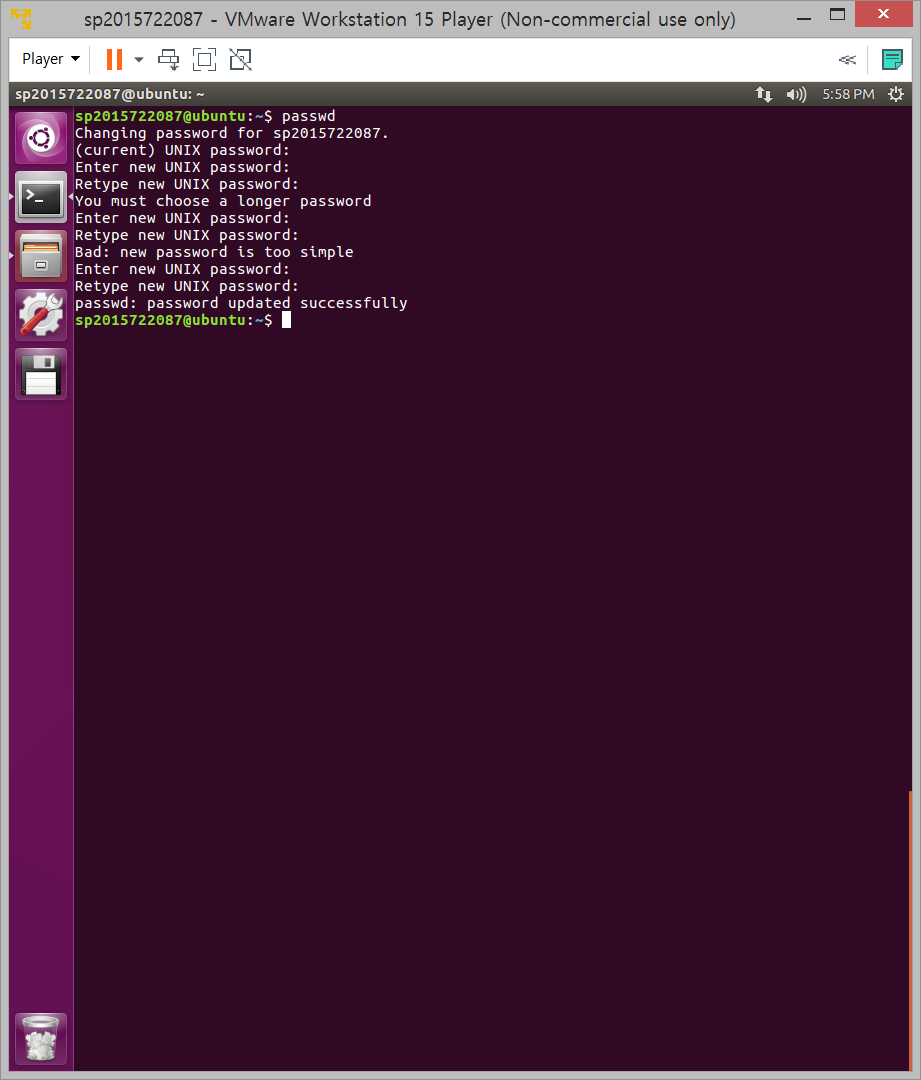
$ ps – 프로세스 상태를 보여주는 명령어이다. 옵션으로 –e를 입력하면 모든 프로세스를 보여주며, -f 옵션을 입력하면 모슨 포맷을 보여준다.



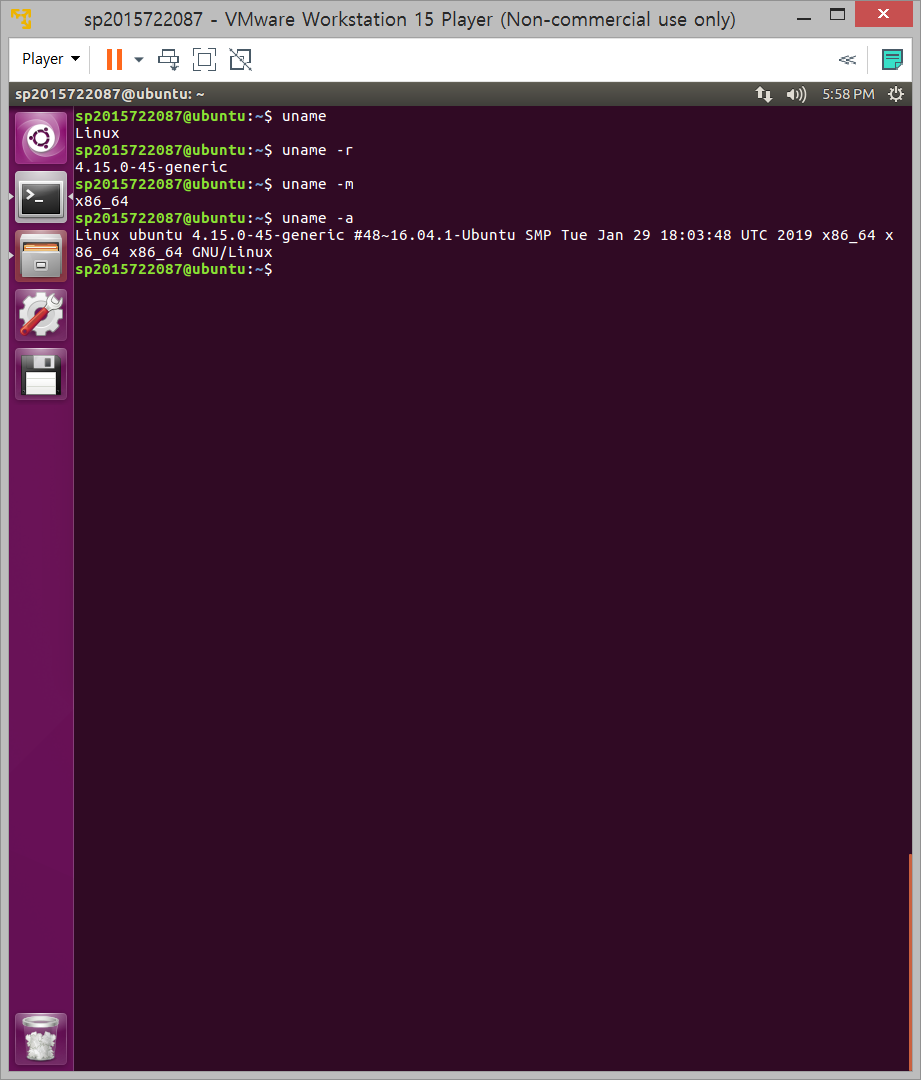
$ pstree – 프로세스 트리를 보여주는 명령어이다. 프로세스간의 연관성을 확인할 수 있다.



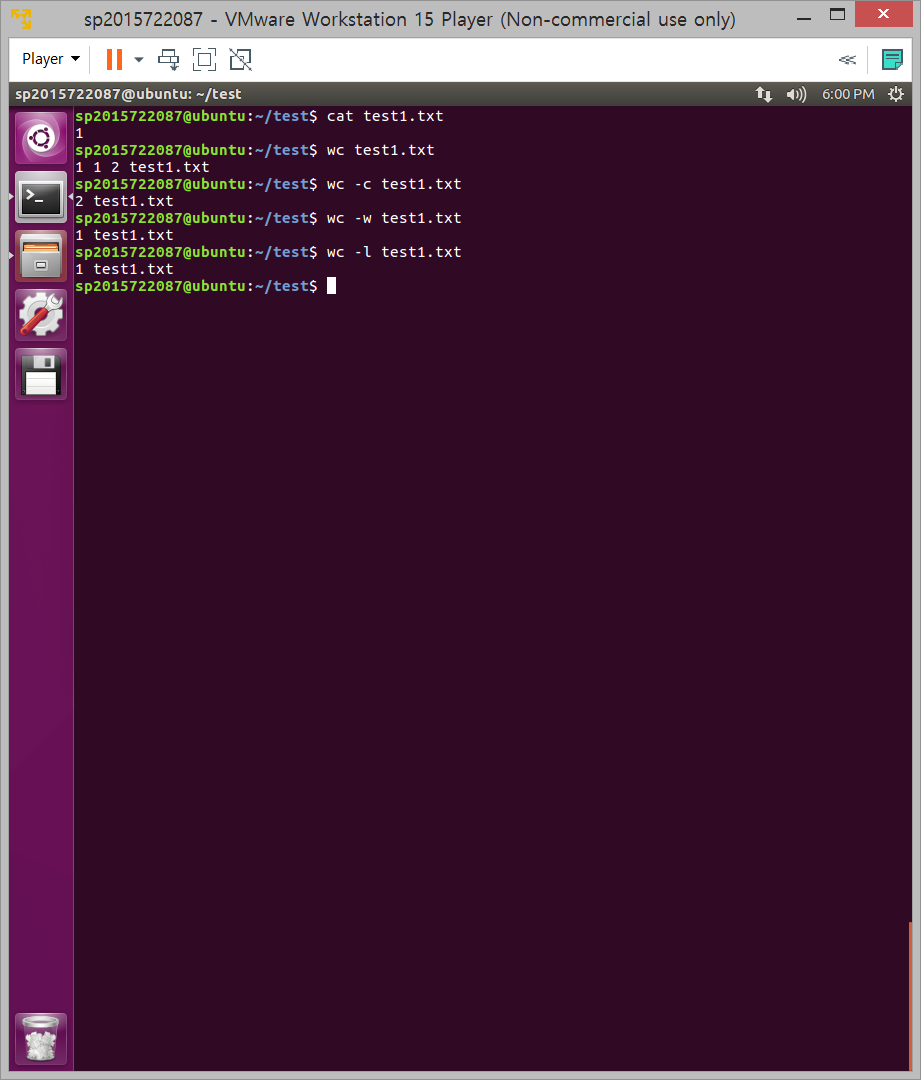
$ time – 프로세스가 실행된 시간을 보여주는 명령어이다. 명령어 뒤에 프로그램 이름을 입력하면 프로그램의 결과와 그 프로그램이 실행된 시간 세가지를 보여준다. 유저 영역에서 실행된 CPU 시간, 커널 영역에서 실행된 CPU 시간. 실제 CPU 시간.



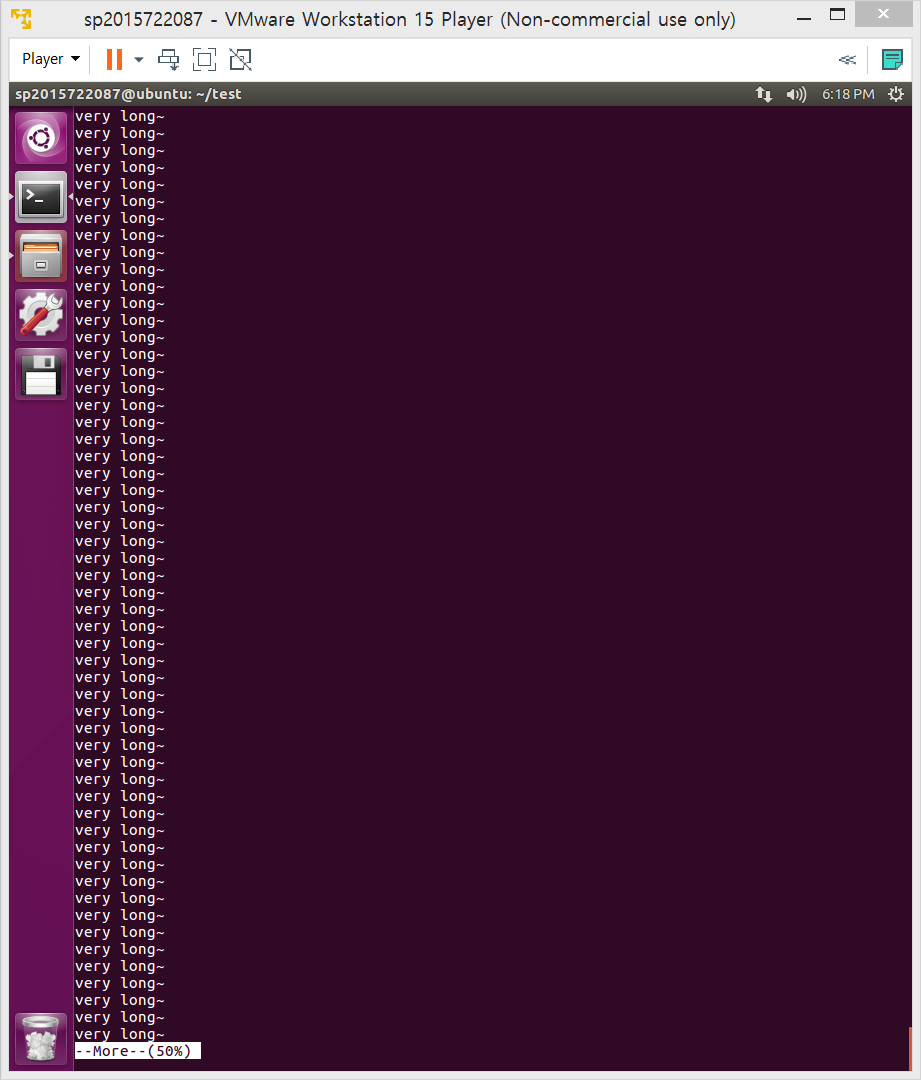
$ passwd – 현재 사용자의 패스워드를 변경하는 명령어이다. 유저들의 패스워드가 저장된 파일은 슈퍼 유저만 읽고 쓰기가 가능한데, passwd 명령어를 이용하여 패스워드를 변경하는 프로그램을 실행하면 이 프로그램은 임시로 권한을 받아 패스워드가 저장된 파일을 수정한다.



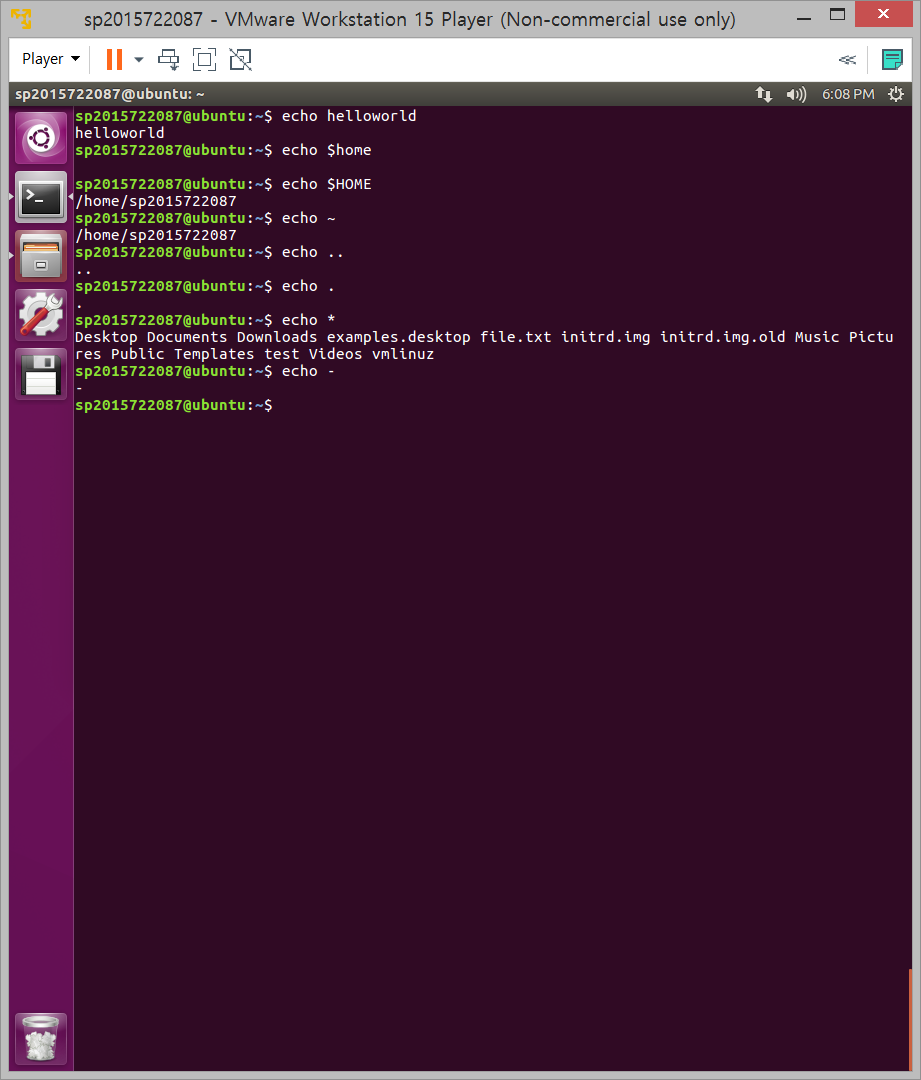
$ uname – 시스템 정보를 출력하는 명령어이다. 옵션을 입력하지않으면 현재 운영체제의 정보 출력하고 옵션으로 –r을 입력하면 커널의 릴리즈를 출력하고 –m을 입력하면 머신 하드웨어의 정보를 –a를 입력하면 모든 정보를 출력한다.



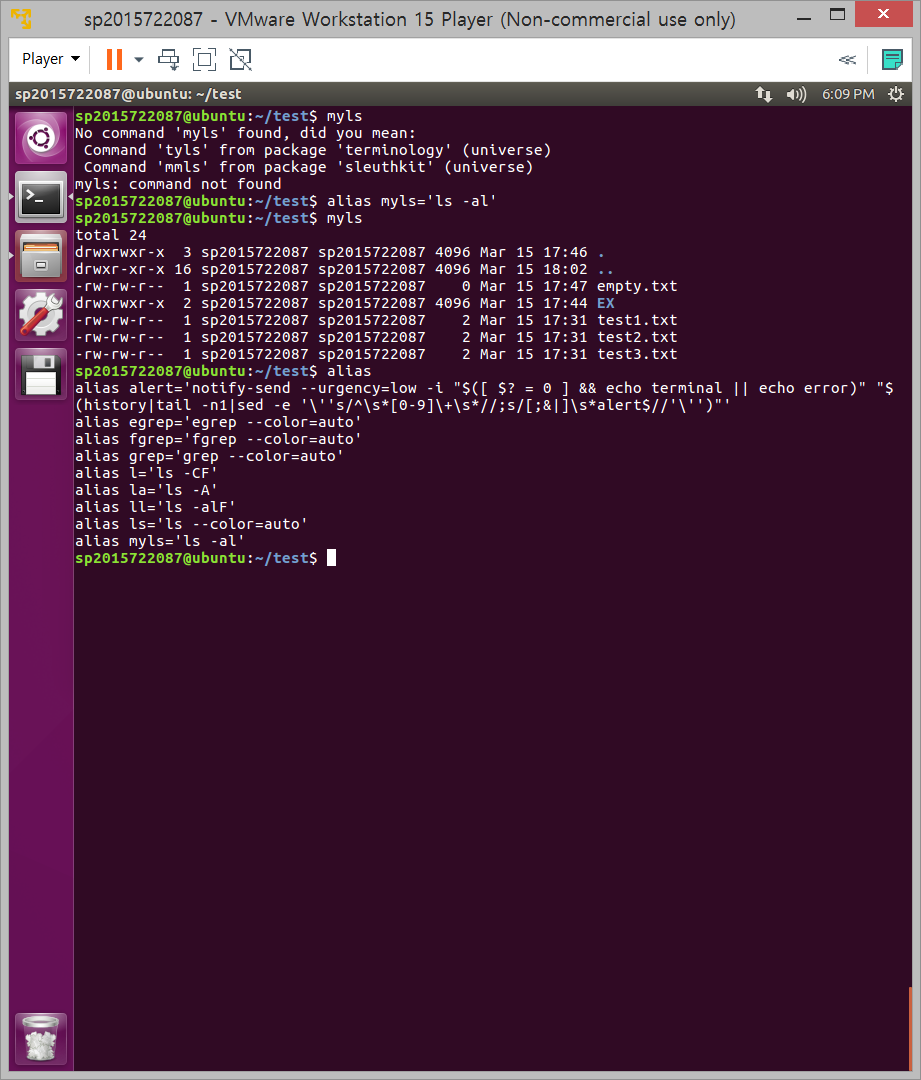
$ wc – 파일의 행, 단어, 바이트 등의 수를 세는 명령어이다. 옵션으로 –c, -w, -l 를 입력해보았는데, -c는 문자의 개수, -w는 단어의 개수, -l는 라인의 개수를 출력한다.



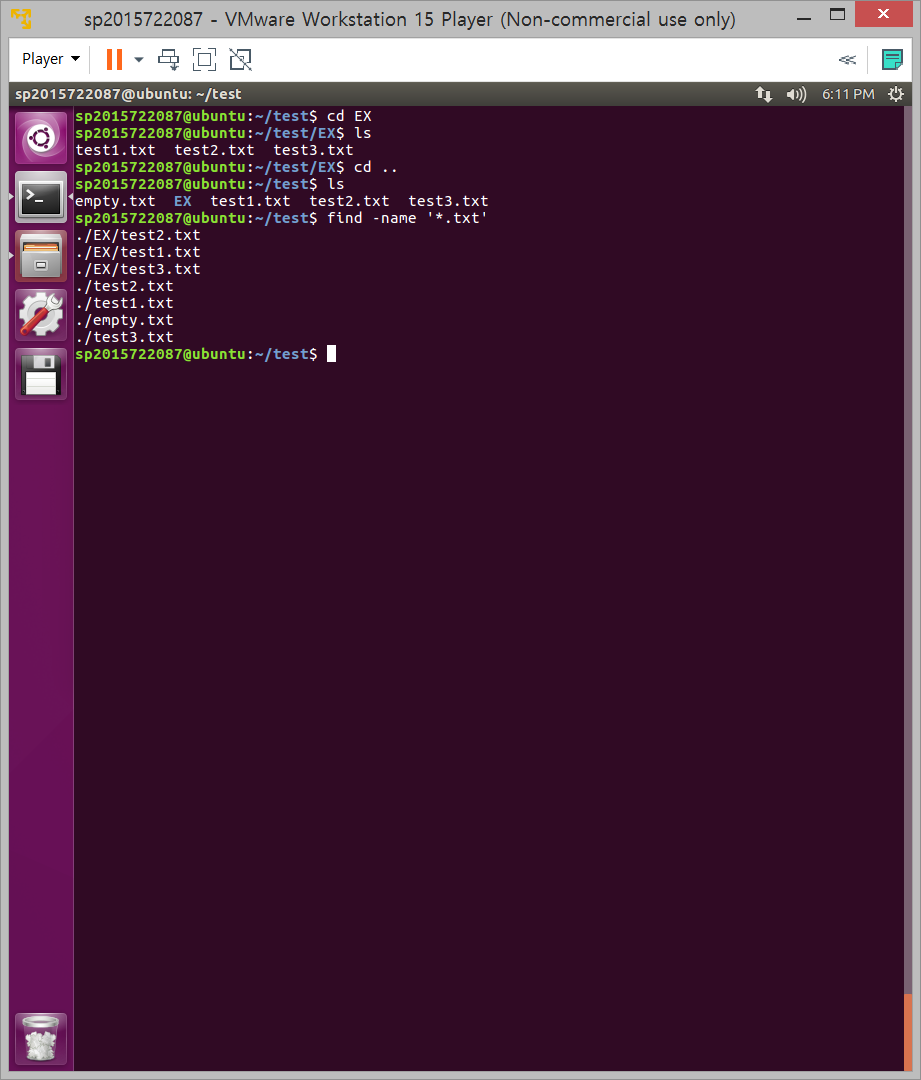
$ more – 파일의 내용을 화면 단위로 끊어서 출력하는 명령어이다. 긴 파일을 확인 할 때는 cat 보다 유용하다는 장점이 있다.



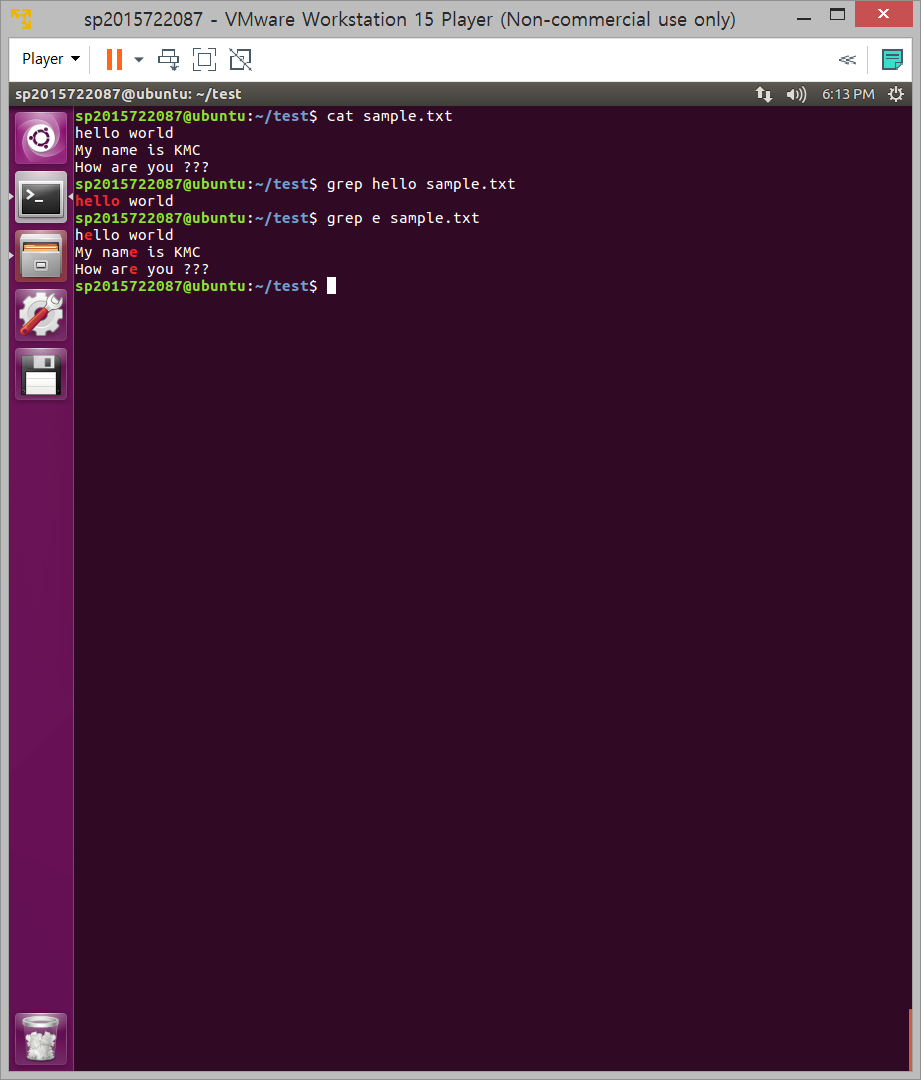
$ echo – 텍스트나 숫자, 또는 디렉토리를 출력하는 명령어이다.



$ alias – 별칭 이라고도 하며, 자주 사용하는 명령어를 특정 문자로 지정하여 간편하게 사용하는 명령어이다. Ls –al 명령어를 myls 로 별칭으로 등록하여 사용해보았다.

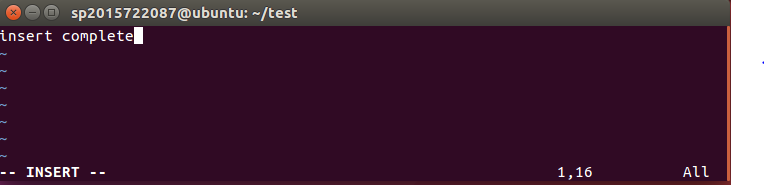


$ find – 원하는 조건의 파일이나 디렉토리를 찾는 명령어이다. Default 는 현재 위치에서 검색이고 특정 디렉토리를 입력하여 특정 디렉토리에서 탐색을 할 수도 있다. –name 옵션을 입력하고 뒤에 찾을 이름을 입력하면 해당 이름이 들어간 파일을 찾아서 디렉토리와 함께 출력한다.

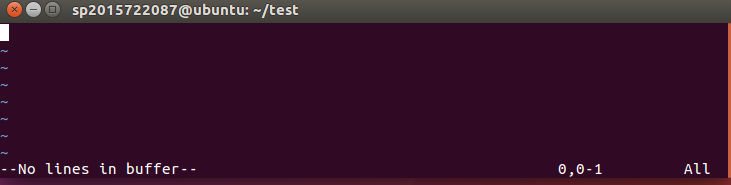


$ grep – 찾고 싶은 키워드를 파일 안에서 찾는 명령어이다. 파일 안에 키워드가 들어간 행을 모두 출력한다.

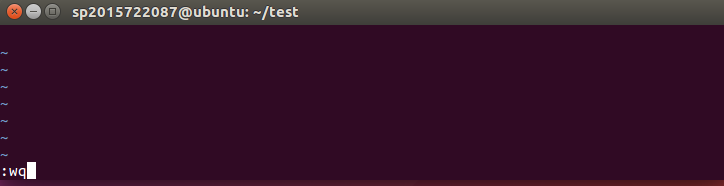
$ vi – vi 편집기를 실행하는 명령어이다. 파일의 내용을 편집하는 Vi 편집기에는 여러가지 명령어가 있는데 명령모드와 입력모드가 있고, 명령모드에서 콜론을 누르면 또 다른 기능을 수행 할 수 있다.



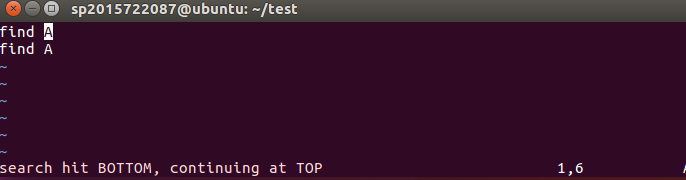
명령모드에서 삽입 명령어 i를 입력하면 아래칸에 –insert—라고 나오면서 입력모드로 들어간다.



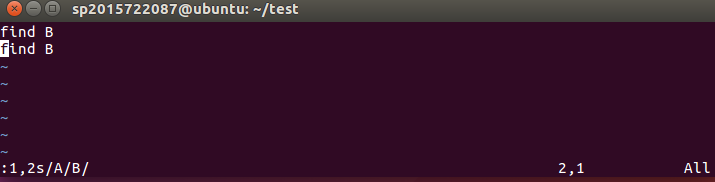
명령모드에서 삭제 명령어 dd를 입력하면 현재 커서위치의 행의 내용을 모두 삭제한다.



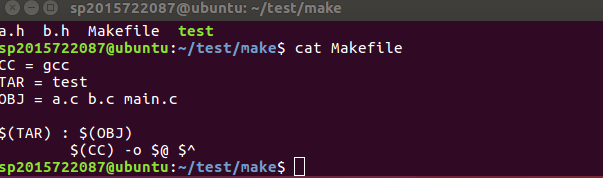
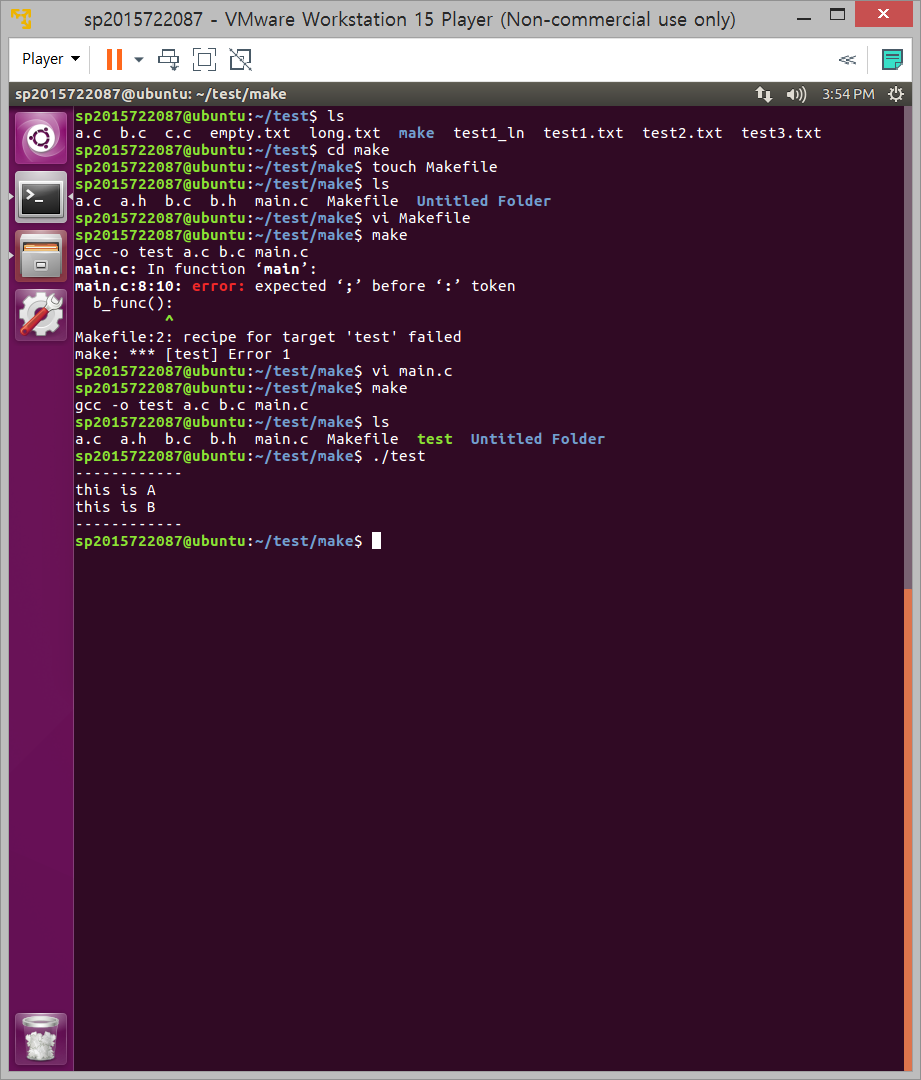
명령모드에서 콜론을 누르고 wq 를 입력하면 데이터를 저장하고 편집기를 종료시킨다.



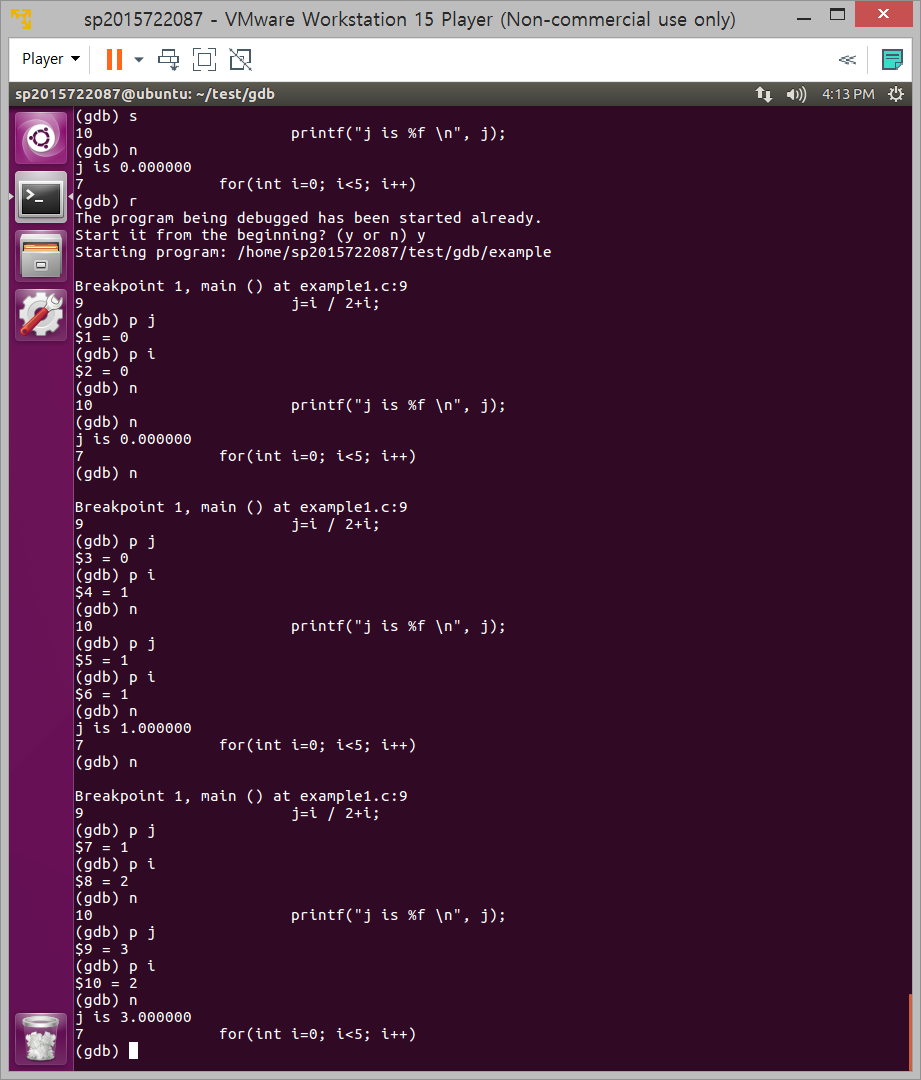
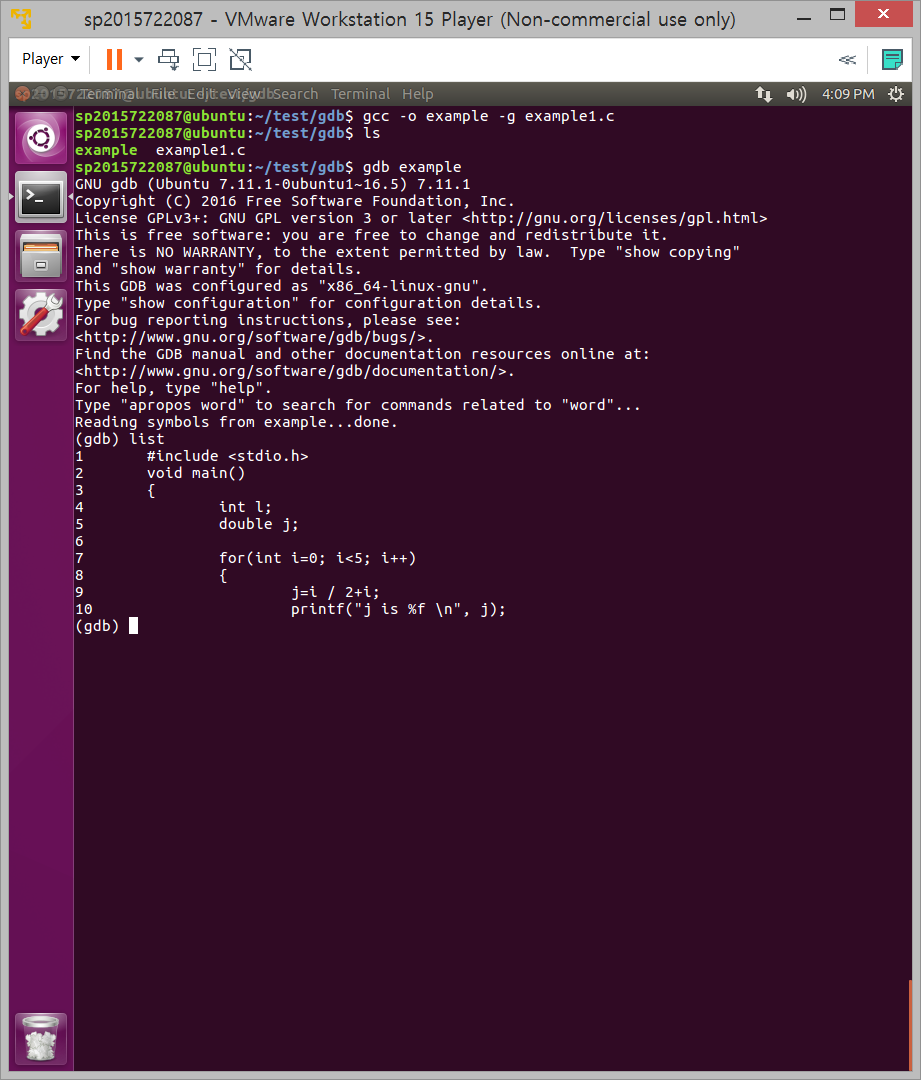
명령모드에서 /(키워드)를 입력하면 현재 커서에서 아래쪽 방향으로 키워드를 찾는다.



명령모드에서 콜론을 입력하고 여러가지 패턴 치환 포맷에 맞게 입력한다. 이곳에서 입력한 방법은 1번째줄에서 2번째 줄까지의 A를 B로 치환하라는 명령이다.



$ make – 컴파일을 자동화하기 위해 사용하는 명령어이다. 매번 컴파일 명령을 일일이 입력하여 하나씩 컴파일한다면 시간이 많이 소요되기 때문에 사용한다. Makefile 에는 컴파일할 때 사용할 소스 파일과 옵션 등에 관해 적어놓은 파일이다. 작업할 대상을 타겟이라고 하고, 콜론 : 뒤에 타겟 명을 입력한다. 다음 줄에는 명령어를 입력하여 실행한다. 여러 개를 하고 싶으면 여러 줄을 작성하면 한번에 컴파일이 된다. 또 자주 사용하는 이름을 변수로 바꾸어 사용할 수도 있다. 타겟은 $@, 타겟명은 $^ 이런 식으로 대체되기도 한다.



$ gdb – 리눅스에서 디버깅을 할 때 사용하는 툴이다. 소스파일을 컴파일하여 생성된 실행파일을 디버깅하여 시작하며, 중간 중간 확인하고 싶은 라인에 breakpoint를 설정해두면 그곳에서 프로그램이 멈추며, 한줄씩 움직이거나 다음 breakpoint로 이동시킬 수 있다. 그때 변수의 정보를 확인하여 프로그램의 어디서 문제가 발생하는지 찾고 수정할 수 있다.

**Reference**

패턴 치환과 검색과 관련된 내용이 강의 자료에 없어 블로그 참조

‘http://igoto.x-y.net/xe/linux/154’ – 패턴 치환과 검색과 관련된 내용을 보고 어떻게 사용하는건지 익힌 후 과제를 해결하는데 사용