1. 자신의 홈 디렉토리 밑에 up 밑에 pr2 디렉토리를 만들고, 그 디렉토리로 이동하라.

   $ cd **$ cd up** **$ pwd (up**여야 함**)**

   $ ls (pr1 존재**) $ mkdir pr2** **$ ls (pr1,pr2** 존재**)**

   $ cd pr2 **$ pwd** (홈**/up/pr2**여야 함)

2. pr2 디렉토리에서 교수님 폴더에 있는 f1 파일을 복사하라. 아래의 pr1, f1에서 1은 소문자 L이 아니고 숫자 1임.

   $ cd ~/up/pr2 (pr2 폴더로 이동; 없으면 위 1번 재실행)

   $ cp /home/jhshim/up/pr1/f1 f1

   $ cp f1 f1.c (f1을 f1.c로 복사함)

   $ ls -l (f1, f1.c가 존재해야 함, -l은 소문자 L)

   $ vi f1 (수업시간에 설명과 함께 실습함. 기다릴 것)

3. vi 명령어를 이용하여 f1.c 라는 파일로 들어가라. 연습은 f1 파일에서 하고, f1.c는 이후의 실습 내용대로 수정하라. 실습 후 f1.c를 검사할 예정임.

   $ cd ~/up/pr2 (pr2 폴더로 이동; 없으면 위1번 실행)

   $ ls (f1.c가 있어야 함; 없을 경우 위 2번 실행)

   $ vi f1.c

[주의사항]

1. **문자 입력 시 갑자기 글자가 깨져 보이거나 커서가 화면의 글자와는 다르게 움직이는 경우가 종종 발생한다. 이 경우 ESC를 한번 누른 후 ctrl+f, ctrl+b를 연속으로 눌러 화면을 새로 디스플레이하면 정상으로 보여짐.**

2. **vi 에디터 내에서 절대 마우스로 오른쪽의 스크롤바를 이용하여 화면을 스크롤시키지 말아야 한다. 실수로 화면을 스크롤시켰다면 ctrl+f, ctrl+b를 연속으로 눌러 화면을 새로 디스플레이하면 정상으로 보여짐.**

4. 위 3번을 실행하고 vi 에디터 내에서 각 행의 번호가 보이게 설정하라.
   :set nu
5. 각 행의 번호가 보이지 않게 설정한 후 다시 보이게 하라.
   :set nonu
   :set nu
6. 탭의 크기를 4로 설정하라.
   :set tabstop=4
7. 자동으로 인덴트가 되게 설정하라.
   :set autoindent
8. 파일 처음(1G)과 끝(G)을 여러 번 반복해서 왔다 갔다 해라. 처음으로 가라.
9. 한 화면 단위로 끝까지 내려 가라. (**ctrl+f** 또는 **PgDn** 반복)
10. 한 화면 단위로 처음으로 올라 가라. (**ctrl+b**또는**PgUp** 반복)
11. 반 화면 단위로 위아래로 옮겨 가라. (**ctrl+d**와 **ctrl+u)**
12. 18번 행으로 커서를 옮겨라. (18G) 그런 후 커서를 현재 라인의 맨 끝으로 옮겨라. (**$ 또는 End)**
13. 커서를 현재 라인의 스페이스가 아닌 첫 번째 글자로 옮겨라.
    정답: **^**
14. 커서를 현재 라인의 맨 처음으로 옮겨라. (숫자**0 또는 Home)** 아마 맨 처음으로 커서가 이동하지 않고 왼쪽으로 더 이상 움직이지 않을 것이다. 커서를 조심스레 좌우로 움직여 보면 printf 앞에 탭 글자가 두 개 있는 것을 확인할 수 있다. 첫 번째 공백(탭) 글자 위치에 커서를 두고 탭을 삭제(x 키 또는 **Delete**키) 하고 그런 후 printf 앞에 있는 공백(탭)에 커서를 두고 다시 한번 삭제(x 키) 하면 두 번째 탭도 사라질 것이다. 이처럼 탭을 삭제할 때는 항상 x 키를 사용한다. 이제 printf 앞에 다시 두 개의 탭을 삽입하라. (p에 커서를 두고 i를 누른 후 탭 키를 두 번 누르고 ESC를 친다.)
15. 파일 처음으로 커서를 옮긴 후 (1G) 찾기로 strtok() 함수를 찾아라. (/strtok [Enter]) 찾은 후 계속 반복하여 strtok()를 찾아라.(n을 반복) (N을 반복해 보라. n과 어떤 차이가 있는가?)
16. 파일 끝으로 가라. (G) 역으로 단어 찾기로 strtok()를 찾아라. (?strtok [Enter]) 찾은 후 계속 반복하여 strtok()를 찾아라.(n을 반복) (N을 반복해 보라. n과 어떤 차이가 있는가?)
17. 파일 처음부터 strtok를 찾아서 모두 strtoken으로 변경하라.
    정답: 파일 처음으로 커서를 옮긴 후 (1G) 찾기로 strtok를 찾는다. (/strtok [Enter]) 찾은 후 strtok의 s 글자 위에서 cw한 후(수정모드) strtoken 입력하여 단어를 변경한 후 ESC를 누른다. 그런 후 n(다음 찾기)과 . (앞의 수정 반복)을 파일 끝까지 반복한다. 파일을 저장한다 (:w)
18. void echo() 함수와 void pwd() 함수의 위치를 서로 바꾸어라.
    정답: echo() 함수 전체의 줄 수를 센다. void 행과 함수 끝의 빈 행도 포함하여 9줄이다. 함수의 첫 행인 void 행(echo 앞의 void)에 커서를 두고 9dd(자르기)한 후 커서를 void rm() 함수의 void 행(rm 앞의 void)으로 끌고 간다. 그런 후 대문자 P(붙여넣기)를 누른다. (잘못된 위치에 복사가 되었으면 u를 누르면 실행취소가 된다. 취소

후 다시 정확한 위치에 커서를 두고 P를 누른다.) 파일을 저장한다 (:w)

19. 기존의 cp() 함수와 유사한 함수 새로운 mv() 함수를 만들고자 한다. 처음부터 새로 만들지 말고 cp()를 복사해서 함수 이름을 mv로 변경하라.
정답: void cp() 함수 전체의 줄 수를 센다. void 행과 함수 끝의 빈 행도 포함하여 23줄이다. 함수의 첫 행인 void 행에 커서를 두고 23yy(복사하기)한 후 커서를 void rm() 함수의 void 행으로 끌고 간다. 그런 후 대문자 P(붙여넣기)를 누른다. 커서를 복사된 cp()의 c 위치로 끌고 간 후 cw-> "mv"입력->ESC를 눌러 함수이름을 mv()로 변경한다. 파일을 저장한다 (:w)

20. mv() 함수 내에서 다음을 실행하라.
1) return; 문장 뒤의 3행을 삭제하라.
(usage: 바로 위의 빈 행으로 커서를 옮긴 후 3dd)
2) 두 개의 goto usage; 문장을 삭제하라. (커서를 해당 문장 위로 옮긴 후 dd로 각 행을 삭제 할 것)
3) If 와 else if 의 두 개의 중괄호 { 와 }를 모두 없애라. ("{"는 x키를 이용해 지우고 "}"는 dd로 삭제한다.)
4) 함수 내의 모든 oldname과 newname 단어를 oldnime, newnime으로 변경하라. (name의 a 글자 위에 커서를 두고 r (한 글자 수정)와 i 키를 연속으로 누른다. 이런 식으로 계속 반복)

21. 찾기 기능으로 cmd_disp[] 배열을 찾은 후 이 배열 초기화 원소로 엔트리 { "move", move }, 를 알파벳 순서에 맞게 적절한 위치에 추가 시켜라.
정답: 찾기 기능 /cmd_disp [Enter]로 찾은 후(cmd_disp_t를 먼저 찾게 됨) 그 아래 쪽에 cmd_disp[] 배열 초기화를 찾아간다. 맨 마지막 항목인 { "rm", rm } 항목으로 커서를 끌고 간 후 yy (복사)한 후 P (붙여넣기) 누른다. 그런 후 위 줄의 { "rm", rm }을 { "move", move }로 변경한다. 이때 단어변경 기능을 사용해서 변경한다. 즉, cw->단어변경->ESC. 두 번째 move 앞의 탭 글자를 x를 이용해 삭제하여 줄을 맞추어라. 파일을 저장한다 (:w)

22. main() 함수를 찾아가라. 찾기 기능(/main[enter])
main() 함수 내에서 다음을 실행하라.
1) 함수의 처음에 int a, b, c;를 삽입하라. (첫 번째 setbuf() 줄에 커서를 옮긴 후 O(빈 줄 삽입) 한 후 int a, b, c;를 입력한 후 엔터를 치고 ESC를 누른다.)
2) 두 개의 setbuf() 함수 사이에 빈 줄을 삽입하라. (첫 번째 setbuf에 커서를 두고 o(소문자 o)를 누른 후 ESC. 삽입된 빈줄에서 dd를 눌러 삭제하고 이번엔 두번째 setbuf에 커서를 두고 O를 누르고 ESC 한다.)
3) for () {}를 한 줄로 모두 합쳐라. (for 행에 커서를 두고 J를 4번 누른다.)
4) 위 for 문장을 원상 복구시켜라. (먼저 {와 printf 사이의 빈 공간에 커서를 두고 i(입력모드)->enter->ESC 누름. 탭을 눌러 인덴트가 들어가게 함. (i->탭키->ESC) 그런 후 ";" 뒤의 빈 공간에 커서를 두고 i->enter->ESC하여 줄을 분리한다. 이를 반복한다. 마지막 } 까지 분리한 후 } 앞의 빈 공간에서 x 키를 누른다. 탭을 하나 삭제하여 for 와 같은 열에 }가 위치하도록 함.
5) proc_cmd();에서 ;을 삭제하라. (커서를 ";" 위로 끌고 가서 x하여 삭제한다.) 이제 이 행의 끝에 "; abc();" 을 새로 추가하라. (a -> "; abc();"입력 -> ESC) 이처럼 행의 끝에 무언가를 추가하려 할 때는 a 키를 사용해야 함. 파일을 저장한다 (:w)

23. 이제 int a, b, c 행의 행 번호를 기억하라.(149행이라 가정) 그런 후 파일을 저장하고 빠져 나와라.
정답: 명령어 모드인 경우 ZZ
삽입모드인 경우 ESC -> ZZ
어떤 모드이든 ESC -> ZZ로 습관 들이도록 하라.

24. 다시 f1.c의 int a, b, c;행(149행이라 가정)으로 바로 찾아가라. 보통 컴파일 에러가 발생한 행으로 바로 찾아 갈 때 이 방법을 자주 사용한다.
정답: $ vi f1.c 한 후 149G
1) 변수 b를 bb로, c를 ccc로 변경하라. (커서를 b 위치로 옮겨간 후 3dw -> i -> "bb, ccc"입력 ->ESC)
2) stdout을 stdin으로, stderr을 stdout으로 변경하라. (stdout의 o 위치로 옮긴 후 cw->in->ESC, 커서를 stderr의 e 위치로 옮긴 후 cw->out->ESC)
3) 파일을 저장한다 (:w)

25. 파일의 맨 끝으로 이동하여 빈 줄을 삽입한 후 "xyz"를 입력한다. (G->o(소문자 o)->[enter]->xyz->[enter]->ESC) 다음의 차이를 실험해 보라.
1) 수정이 된 상태에서 **:q[Enter]** 입력하면 저장되지 않았다고 에러가 나올 것이다.
2) 저장하고 빠져 나오기인 :wq[Enter]하라. ZZ와 동일

한 기능임. 다시 vi f1.c로 들어간다. xyz 행을 찾아라. (/xyz[enter])

3) 찾았다면 xyz행을 삭제한 후 수정취소하기를 하라. (dd 한 후 즉시 u를 누른다.) xyz 뒤에 def를 추가, 커서를 xyz 앞에 끌고 간 뒤 ABC를 추가, ESC 한 후 U를 누름 (한 줄 전체 변경 취소). 다시 xyz 뒤에 def를 추가, 커서를 xyz 앞에 끌고 간 뒤 ABC를 추가, ESC 한 후 이번엔 u를 누름 (마지막 수정인 ABC 추가한 것만 취소됨)

4) :q![Enter]로 수정한 내용을 저장하지 않고 vi 종료한다. 만약 vi에 들어간 후 어떤 내용도 수정하지 않았거나 또는 :w[enter]를 이용하여 저장한 후 그 이후에 어떤 내용도 수정하지 않은 상태에서 그냥 vi 빠져 나오기를 할 경우에는 :q[enter]를 사용할 수 있다. 다시 들어가 마지막(G) xyz 행을 지우고(dd) 저장(:w)한 후 빠져 나오라 (:q).

26. 종종 파일을 저장하기 위해 ZZ을 누른다는 것이 실수로 ctrl+Z을 누르는 경우가 있다. vi f1.c로 들어가라.

1) ZZ을 누르면 정상적으로 저장하고 빠져 나온다.
vi f1.c로 다시 들어가라.

2) 이번엔 ctrl+Z를 눌러보라.
"중단됨 (사용자)"
메시지가 나온 후 명령 프롬프트가 나올 것이다. 이 상태는 정상 종료된 것이 아니라, vi 에디터 프로그램의 실행을 일시적으로 중단시키고 빠져 나온 것이다. 만약 파일을 수정하였다면 수정했던 내용이 저장되지 않은 상태이다. 그러나 프로그램이 죽은 것은 아니다. 시스템 내에 살아 있다.
$ ps [enter]
현재 내가 실행시킨 프로그램 목록을 보는 명령어인데 bash, ps, vi 등 세 개의 프로그램이 보여질 것이다.
또는
$ jobs [enter]
중단되었거나 백그라운드로 실행되는 프로그램을 볼 수 있다. vi f1.c 가 중단되었다는 표시가 될 것이다. 만약 중단된 vi가 여러 개 있다면 각각의 job(vi)을 모두 보여 주고 앞에 job의 번호가 보여 질 것이다.
다시 vi 프로그램의 실행을 복구하려면
$ %1 [enter] // %1은 job(vi)의 번호임
다시 vi 에디터로 복구 되었다. :q로 vi 에디터 빠져 나간다. 만약 중단된 vi가 여러 개 있다면 %?하여 복구하라. ?는 복구할 job의 번호임.
$ ps [enter] 또는 **$ jobs [enter]**
vi 프로그램이 보이지 않을 것이다.

3) 결론적으로 vi 에디터 내에서 실수로 ctrl+Z를 눌러 중단된 상태에서 빠져 나왔다면 $ %1 로 실행을 복구시키면 된다.

4) 앞으로 이런 경우 반드시 경험하게 되므로 꼭 기억하기 바란다.

27. vi 에디터 내에서

1) 문자 입력 시 갑자기 글자가 깨져 보이거나

2) 커서가 화면의 글자와는 다르게 움직이는 경우가 종종 발생한다.

3) 한 줄의 길이가 명령 창의 화면 크기보다 더 길어질 경우에도 종종 이러한 현상이 발생함

이 경우 ESC를 한번 누른 후 ctrl+f, ctrl+b를 연속으로 눌러 화면을 새로 디스플레이하면 정상으로 보여진다.

28. 다음의 프로그램을 실행하여 정상적으로 실습했는지 확인하기 바란다. 아래 2는 실습 번호임.
$ /home/jhshim/util/fshw 2
매번 [enter]하기 전에 "**0** 개의 부분 오류. 계속[enter]?" 가 나오는데 숫자가 0이어야 정상이다. 틀린 개수와 틀린 이유가 표시된다. 끝날 때까지 계속 enter를 눌러라. 디렉토리 또는 체크하고자 하는 파일이 존재하는지, 그리고 소스파일인 경우 변수, 함수 등등이 제대로 구현되어 있는지 확인한다.

1) 어떤 프로그램 또는 명령어든 실행 중에 끝내려고 한다면 (위 fshw도 마찬가지) 무조건 **Ctrl+C** 를 누르면 종료된다.

또 다른 프로그램인 eshw도 활용해 보자. 아래 2는 실습 번호임.
$ /home/jhshim/util/eshw 2