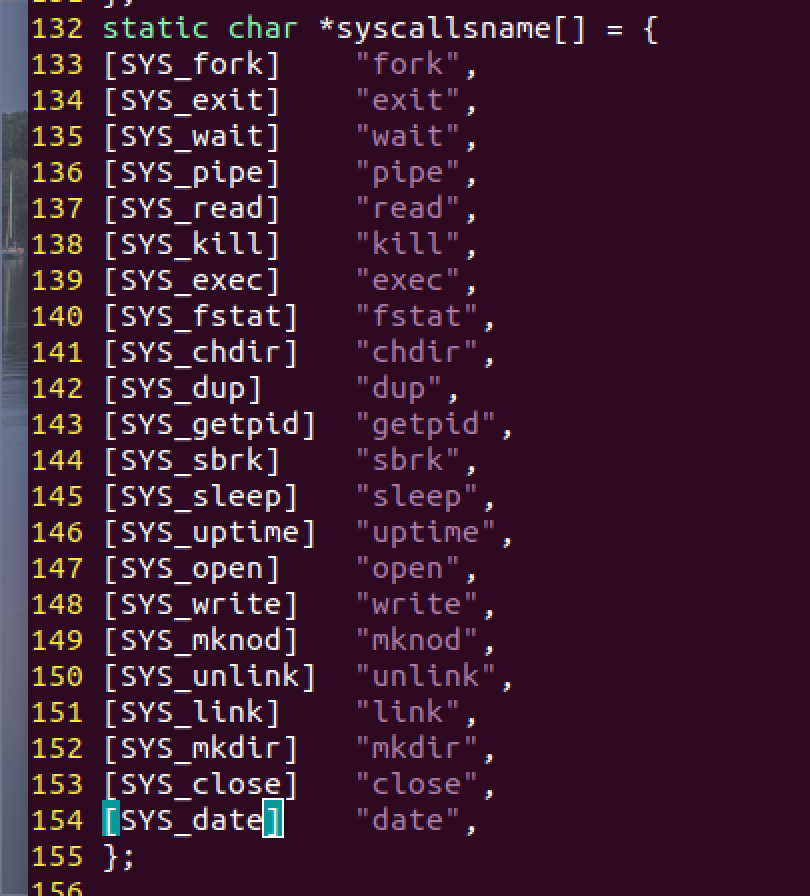
**System Programming Assignment 3**

2015005205

최홍규

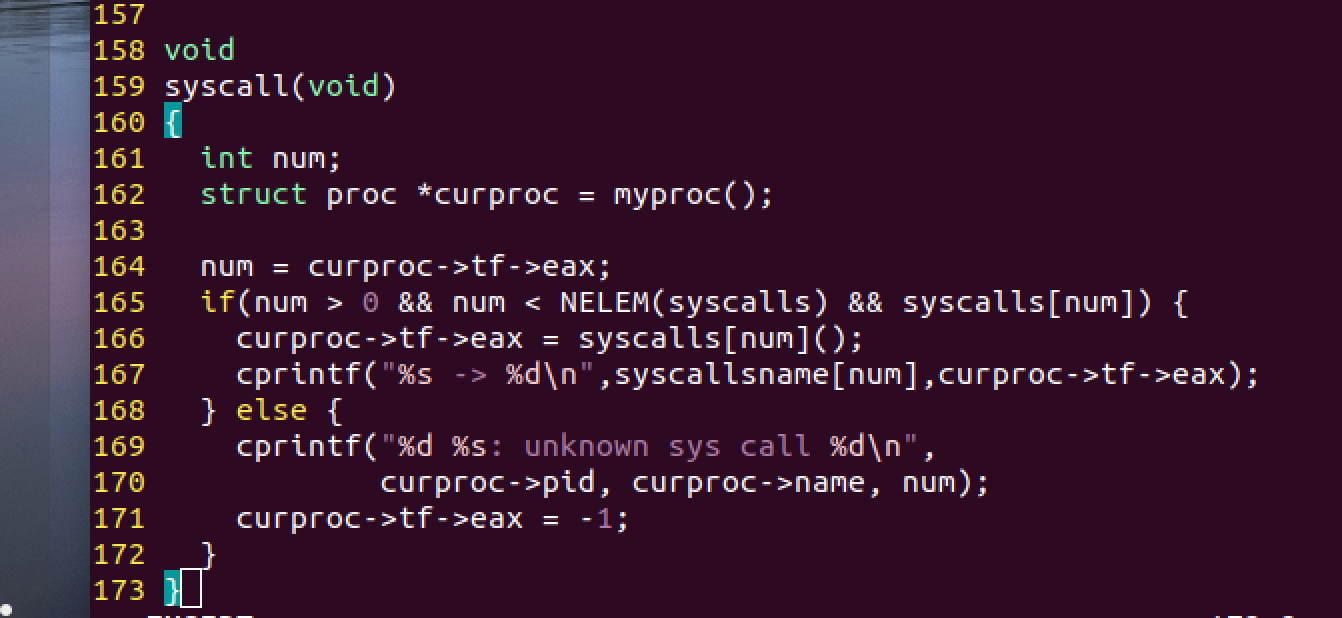
**Part One : System call tracing**



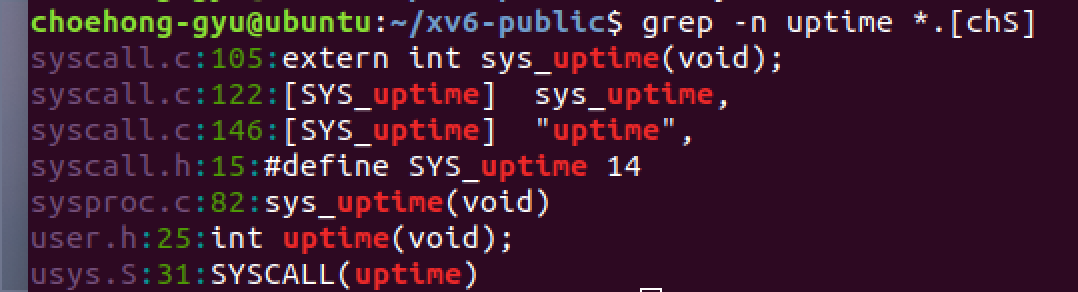
-syscalllname이라는 문자열 배열에 systemcall 번호에 맞는 위치에 각 systemcall함수 이름을 넣어주었다.

-syscall함수에서는 현재 실행된 systemcall 함수의 번호에 맞는 systemcall함수 이름을 문자열 배열을 통해 출력한다.

- curproc->tf->eax에 systemcall의 반환값이 저장된다.

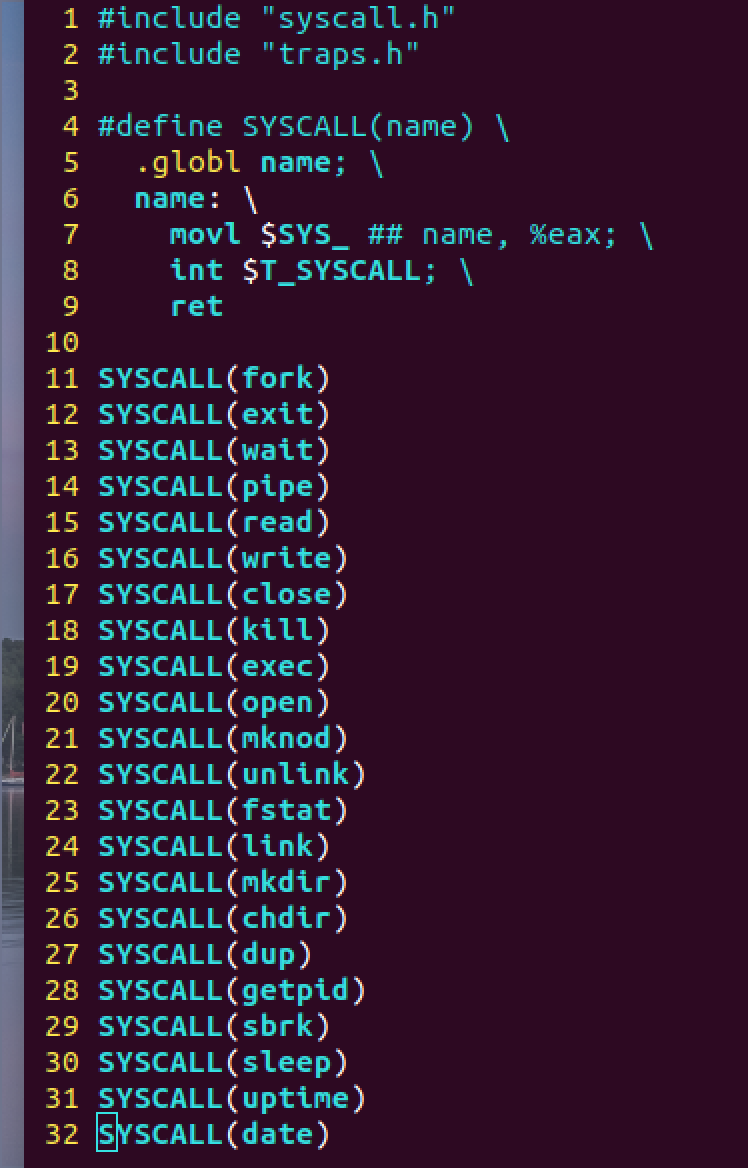


**Part Two : Date system call**

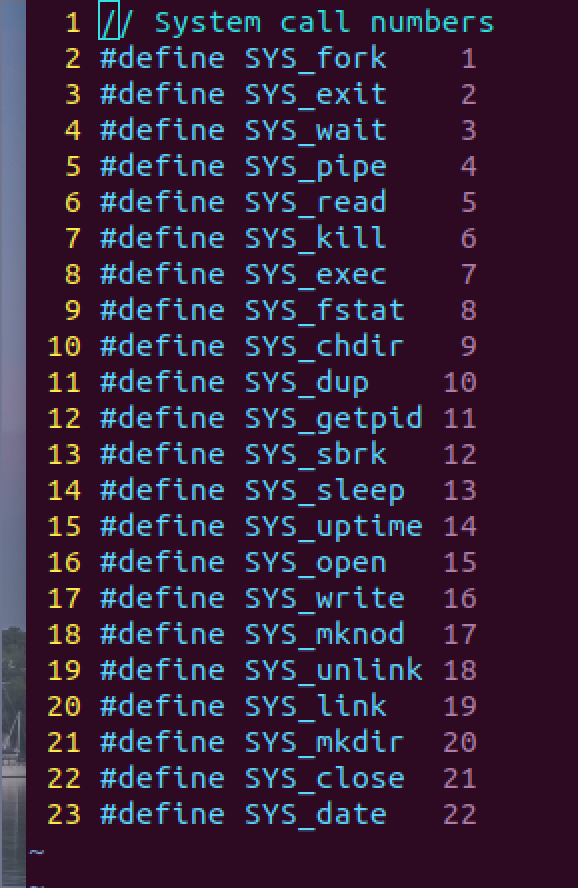


**-grep -n uptime \*.[chS] 결과.**

Date시스템 콜을 사용하기 위해 미리 만들어져 있는 시스템 콜이 쓰이는 부분을 확인해 보았다.



<-usys.S

eax레지스터에 시스템 콜 번호를 저장한 후 int instruction 수행한다.

syscall.H->

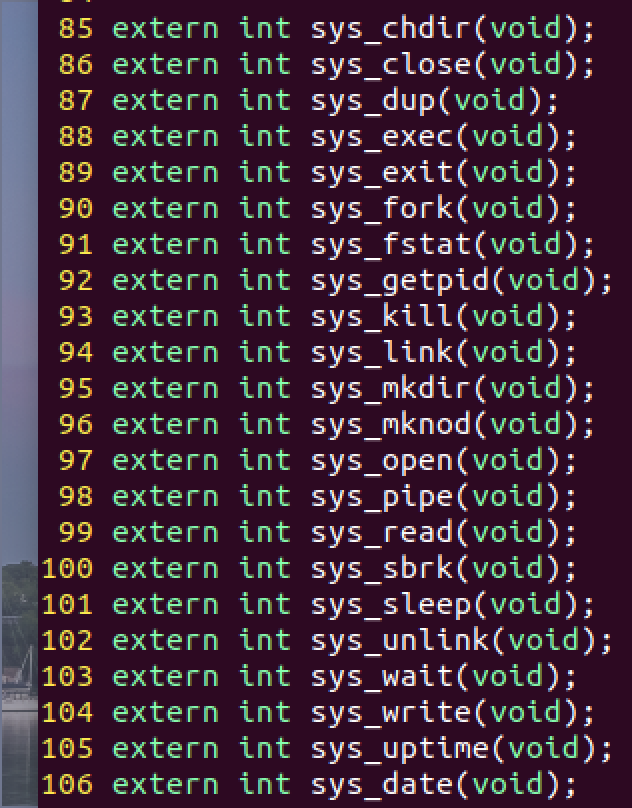
시스템 콜에 해당하는 번호가 선언

syscall 함수에서 시스템 콜을 식별하기 위한 식별자로 사용된다.

date시스템 콜의 번호를 22로 지정해 주었다.

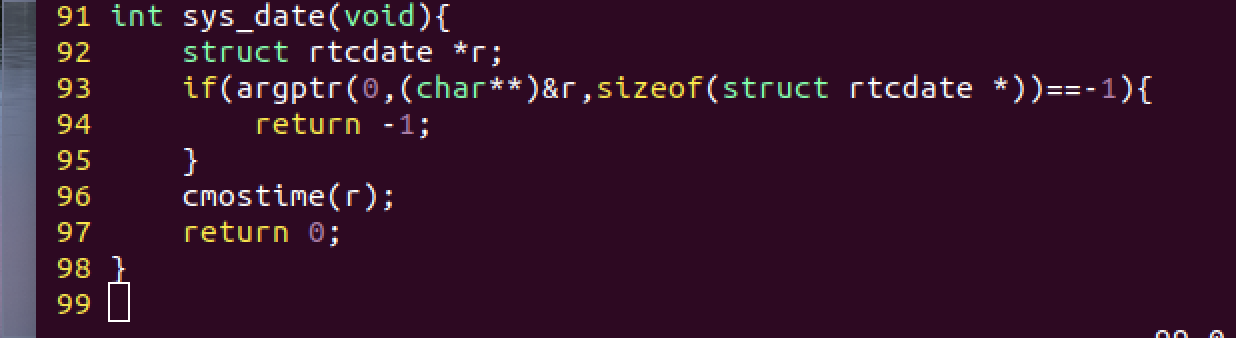
<- syscall.c

함수포인터 배열에 시스템콜 핸들러를 저장해준다. date시스템콜의 핸들러도 함수포인터 배열에 추가해 주었다.



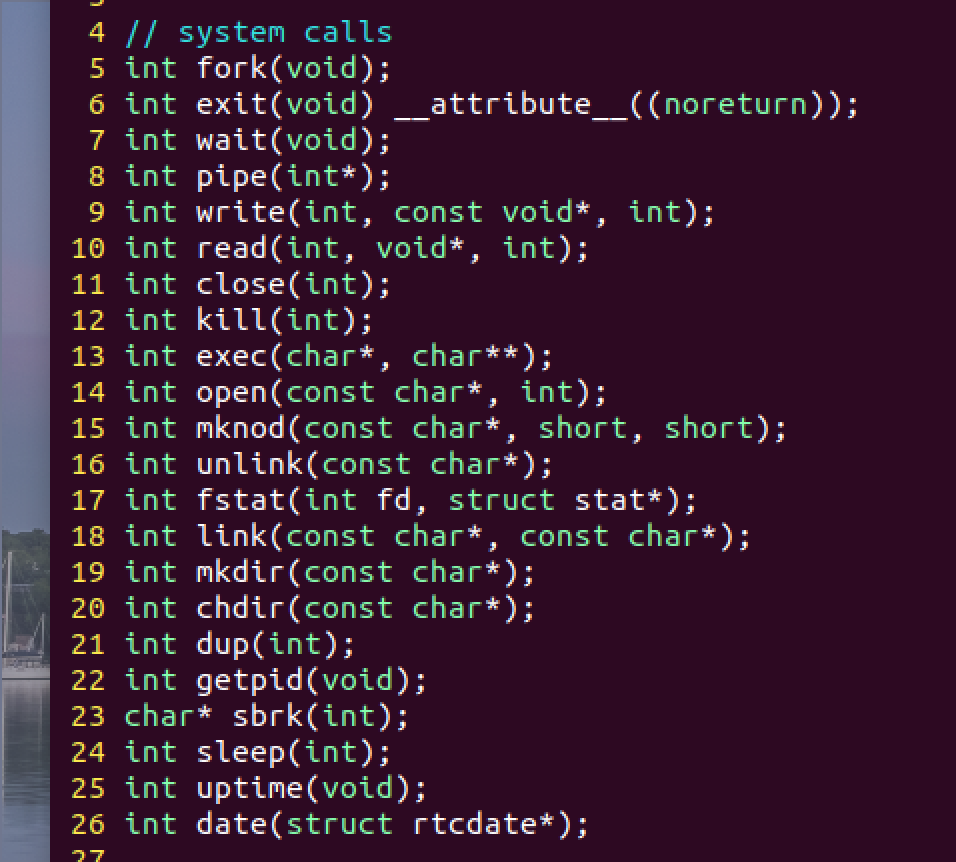
syscall.c->

시스템콜 핸들러 함수들을 다른 곳에서 사용할 수 있도록 extern int 함수로 선언해 주었다.

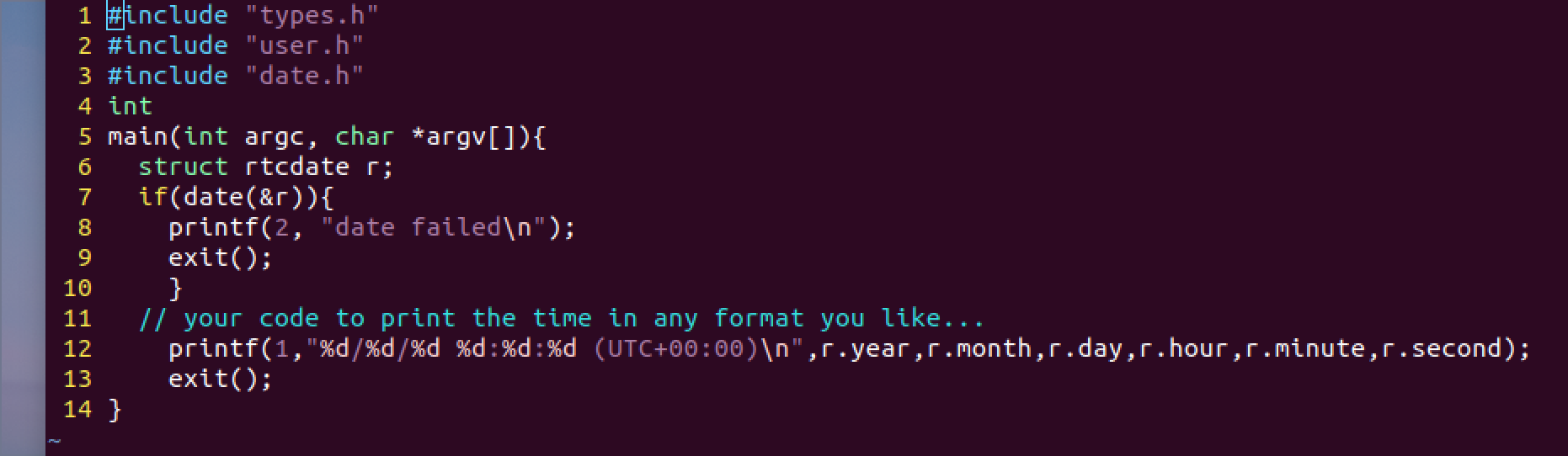


-sysproc.c

시스템 콜 핸들러 구현 부분이다. 시스템 콜 호출할 때 매개변수로 들어온 포인터의 주소를 rtcdate 구조체에 저장한 후 cmostime함수를 호출한다.

<- user.h

파일에 시스템 콜 목록을 저장해 두었다. date시스템 콜을 추가해 주었다.



-date.c

shell에서 입력으로 date가 들어오면 실행되는 코드이다. 시스템 콜 date를 실행한 후 현재 시간을 형식에 맞게 출력해준다.