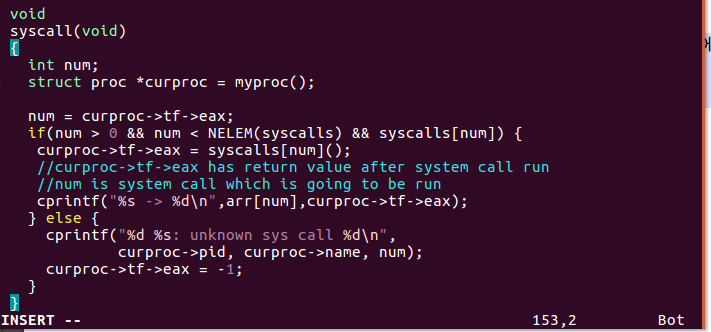
3-1)



//수정된 syscall 함수

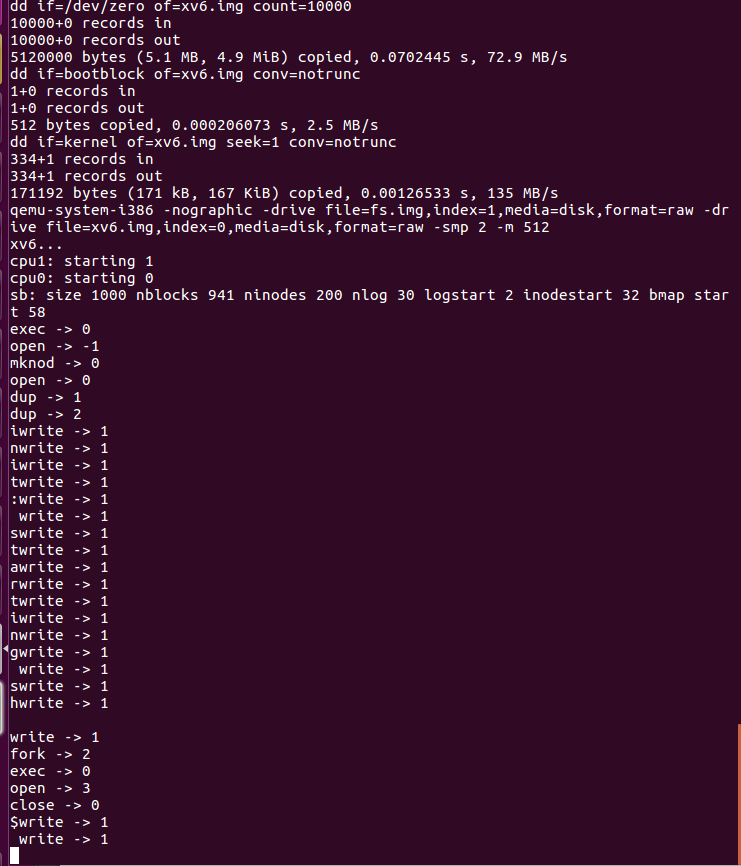


지금 현재 실행할 system call함수 매크로 넘버는 curproc->tf->eax에 저장되어있고 num에 담겨있다.

Proc.h에 들어가보면 tf가 현재 syscall의 trap frame임을 알 수 있다.

그 system call 매크로 번호에 맞게 text 배열을 만들어서 출력하도록 하였다.

Xv6 부팅시 출력되는 system call과 return 값



3-2)

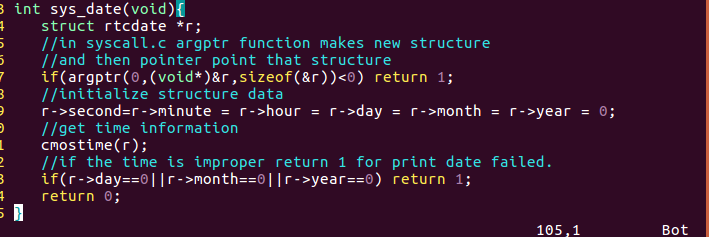
System call 함수들이 구현되어 있는 sysproc.c 파일에 새로운 system call 함수인 date 함수를 구현하였다.

우선 타입에 맞는 rtcdate structure 포인터를 선언하고

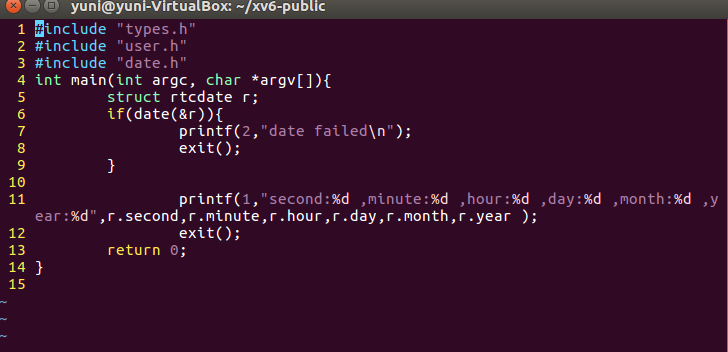
다른 system call 함수들이 사용하는 argptr 함수를 사용해서 structure를 만들어 포인터가 가리키게 한다. 그리고 0값들로 초기화후 cmostime함수를 사용해서 시간을 받아온다.

만약 structure가 잘 만들어지지 않았거나 시간 값이 제대로 안 들어 갔으면 return 1 > date failed 출력함

//sysproc.c에 구현된 date함수



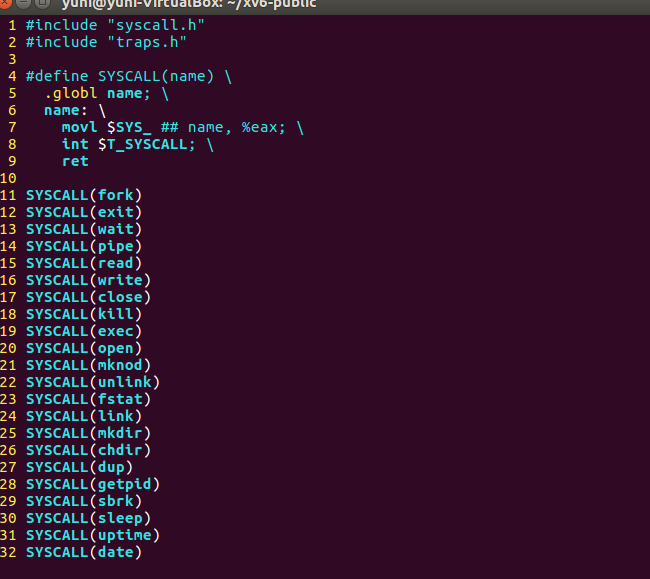
//date.c에 추가된 printf문



System call을 다루기 위한 처리

// usys.S에 date 추가

SYSCALL 이름을 확인하고 맞게 %eax로 올리도록



//Makefile에 추가된 date system call

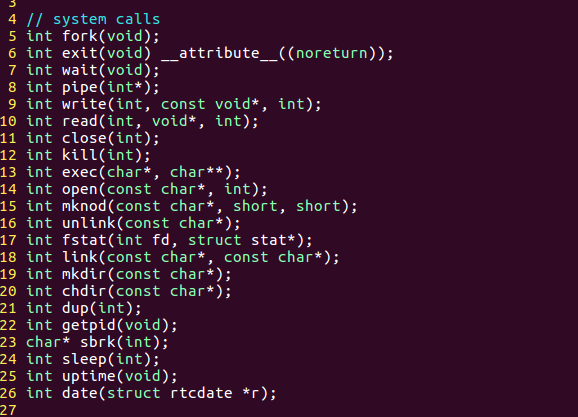
Make 할 수 있도록



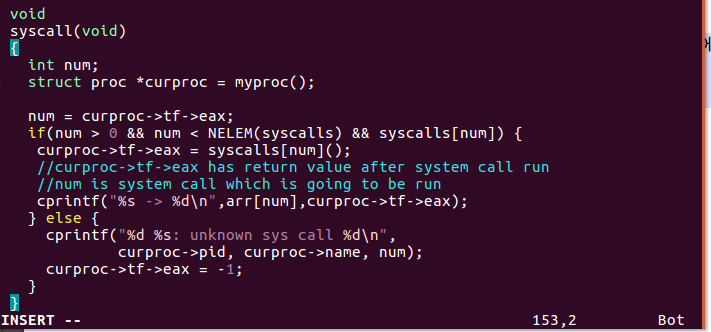
//syscall.h 에 추가



// user.h 에도 추가 user 프로그램에서도 알수있게

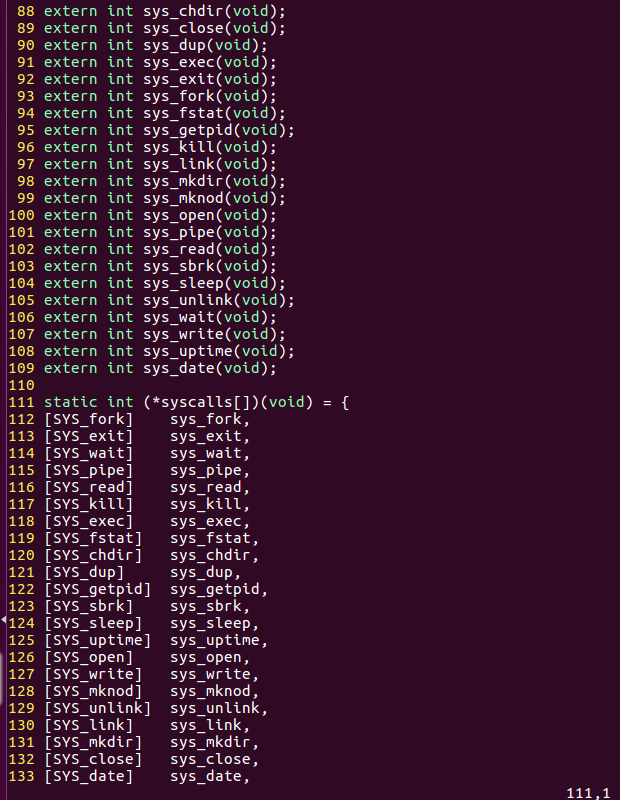


Syscall 받고 syscalls[num]();에서 실행>>



//syscall.c에 추가

Syscall.h에 SYS\_date 정의하고 syscall.c에 추가해서 date function을 system call로 등록한다.



date 명령 입력이후의 출력

