

Homework 3: xv6 system calls

제출마감 : 2018.10.11 11:00 am

제출처 : 손인엽 조교(son901217@naver.com)

Part One: System call tracing

첫 번째 과제는 시스템 콜이 실행될 때마다 특정 출력을 수행하도록 xv6 코드를 수정하는 것이다. 수행된 시스템 콜의 이름과 반환 값을 출력한다.

수정이 완료되면, xv6 가 부팅 될 때 아래와 같은 출력을 볼 수 있다.

```
...
fork -> 2
exec -> 0
open -> 3
close -> 0
$write -> 1
write -> 1
```

xv6 가 부팅 될 때 shell 을 fork 하고 exec 를 통해 실행시킨다. Shell 은 두 개의 file descriptor 가 오픈 된 것을 확인하고 \$ 프롬프트를 출력한다.

(Note: shell 은 write 시스템 콜을 사용하여 출력문을 출력하기 때문에 쉘의 출력문과 시스템 콜의 출력문이 섞인다.)

Hint: syscall.c의 syscall() 함수를 수정하라.

submit: 수정한 syscall() 함수

Part Two: Date system call

두 번째 과제는 xv6 에 새 시스템 콜을 추가하는 것이다. 이 과제를 통해 시스템 콜 메커니즘을 확인할 수 있다.

새로 구현할 시스템 콜은 다음과 같다. 현재 UTC 시간을 가져와 유저 프로그램에게 반환해야한다. 현재 UTC 시간을 읽기 위해서는 lapic.c 에 정의되어 있는 cmostime() 함수를 사용해야 한다. 해당함수에서 사용하는 매개변수인 struct rtcdata 구조체는 data.h 파일에 정의되어 있다.

이번 과제에서 생성할 date 시스템 콜을 호출하는 유저 프로그램을 만들어야 한다. 아래는 새로 추가할 할 유저 프로그램 date.c 의 코드이다. Xv6 디렉터리에 해당 소스파일을 만들어서 저장하라.

```

#include "types.h"
#include "user.h"
#include "date.h"

int
main(int argc, char *argv[])
{
    struct rtcdate r;

    if (date(&r)) {
        printf(2, "date failed\n");
        exit();
    }

    // your code to print the time in any format you like...

    exit();
}

```

새 `date` 프로그램이 `xv6` 에서 동작하게 하기 위해서는 `Makefile` 에 정의되어 있는 `UPROGS`에 `_date`를 추가해야 한다.

새로운 시스템 콜을 만들기 위해 미리 만들어져 있는 시스템 콜과 관련된 코드들을 참고할 수 있다. 예를 들어, “`uptime`” 시스템 콜이라면 “`$ grep -n uptime *.c`” 명령어를 통해 모든 파일 중에서 `uptime`을 사용하는 코드를 찾을 수 있다.

구현이 완료 되었다면, `xv6 shell` 에서 `date` 를 입력하면 현재 UTC 시간이 출력될 것이다. 수정된 각각의 파일에서 `date` 시스템 콜이 진행되는 과정을 설명하라.

submit : `date` 시스템 콜을 생성하기 위해 수정 및 추가 된 부분(파일 명과 파일 내에서 수정되거나 생성한 코드)

제출양식 : part 1, 2 내용을 “`hw3_학번`” 문서로 작성하여 제출