Homework: xv6 CPU alarm

제출마감 : 2018.10.18. 11:00am

제출처 : 손인엽 조교(son901217@naver.com)

이번 과제를 통해 xv6 에 새로운 기능을 추가하게 될 것이다. 구현 해야하는 기능은 프로세스가 CPU 를 사용할 때 주기적으로 프로세스에 알려주는 기능이다. 이 기능은 프로세스가 얼마나 CPU 시간을 사용했는지 확인하는데 유용하며, 일정 시간이 지났을 때 특정 코드를 수행하게 하는데에도 유용하다.

먼저 `alarm(interval, handler)` 시스템콜을 구현한다. 프로세스가 `alarm(n, fn)` 함수를 호출하면, 이후 n번의 “tick”마다 해당 프로그램에 구현되어있는 fn 함수를 수행하도록 구현하여야한다. fn 함수가 끝나면, 원래 수행되던 프로세스의 상태로 돌아와야 한다. tick은 타이머 하드웨어에 의해 생성되는 주기적인 시간 단위이다.

시스템콜 구현이 끝나면 `alarmtest.c` 파일을 만들어 아래 코드를 삽입한다.

```
#include "types.h"
#include "stat.h"
#include "user.h"

void periodic();

int
main(int argc, char *argv[])
{
  int i;
  printf(1, "alarmtest starting\n");
  alarm(10, periodic);
  for(i = 0; i < 25*500000; i++){
    if((i % 250000) == 0)
      write(2, ".", 1);
  }
  exit();
}

void
periodic()
{
  printf(1, "alarm!\n");
}
```

위의 프로그램은 `alarm(10, periodic)` 을 호출하여 kernel 이 매 10 tick마다 periodic 함수를 수행하게 한다. alarm 시스템콜을 제대로 구현했다면, alarmtest 프로그램을 수행하였을 때 다음과 같은 출력이 나와야 한다.

```
$ alarmtest
alarmtest starting
.....alarm!
....alarm!
.....alarm!
......alarm!
.....alarm!
....alarm!
....alarm!
......alarm!
.....alarm!
...alarm!
...$
```

(만약 alarm! 메시지가 한번밖에 뜨지 않는다면, alarmtest.c 파일의 for 문의 반복량을 10배로 늘리면 된다.)

힌트1: `alarmtest.c` 를 xv6 유저프로그램으로 컴파일하기 위해 Makefile 을 수정해야한다.

힌트2: `user.h` 에 추가해야 할 선언은 아래와 같다.

```
int alarm(int ticks, void (*handler)());
```

힌트3: 아래는 `sys_alarm()` 함수의 코드이다.

```
int
sys_alarm(void)
{
  int ticks;
  void (*handler)();

  if(argint(0, &ticks) < 0)
    return -1;
  if(argptr(1, (char**)&handler, 1) < 0)
    return -1;
  myproc()->alarmticks = ticks;
  myproc()->alarmhandler = handler;
  return 0;
}
```

힌트4: `sys_alarm()`은 알람 간격과 핸들러 함수의 포인터를 `struct proc` 구조체에 새로운 필드로 저장해야 한다. `proc.h` 를 확인하라.

힌트5: 프로세스의 알람 핸들러 함수를 마지막으로 호출하고 난 몇번의 tick 이 지났는지 기록해야 한다. 이는 `struct proc` 구조체에 새로운 필드를 만들어 기록하면 된다. `struct proc` 의 필드들은 `proc.c` 파일의 `allocproc()` 함수에서 초기화된다.

힌트6: 매 tick 마다 하드웨어 클럭은 인터럽트를 발생시키고 인터럽트는 `trap()` 함수의 `case T_IRQ0 + IRQ_TIMER` 에서 관리된다.

힌트7: 프로세스가 실행 중이고, 유저 프로세스에서 발생한 타이머 인터럽트일 때에만 알람 tick 을 다루고 싶다면 아래와 같이 코딩하면 된다.

```
if(myproc() != 0 && (tf->cs & 3) == 3) ...
```

힌트8: `IRQ_TIMER` 핸들러에서 프로세스의 알람 시간에 다다르면 알람 핸들러를 실행해야 한다.

힌트9: 핸들러에서 리턴되면 프로세스가 중단된 시점부터 다시 실행되도록 해야 한다.

Submit: 수정한 trap.c 파일

제출양식: hw5_자신의학번.c