

## 1. 개요

xv6에 두 시스템 콜 date와 alarm을 추가하고, 각각의 시스템 콜을 사용하는 예제 프로그램을 구현한다. date는 현재 시스템의 시간을 구조체 rtcdate에 저장하는 시스템 콜이며, alarm은 n초 후에 현재 프로그램을 종료시키는 시스템 콜이다. 구체적으로 date는 cmostime 함수를 사용하여 동작하며, alarm은 이보다 복잡한데, alarm의 timer 관련 정보를 저장하기 위해 구조체 proc에 필드를 추가하고, trap 함수에서 alarm의 tick을 처리하여 구현된다.

## 2. 설계

시스템 콜 추가는 user.h, usys.S, syscall.h, syscall.c, sysproc.c의 파일을 수정한다.

### 2-1. 소스코드별 함수 개요

#### 2-1-1. alarmtest.c / datetest.c

추가한 시스템 콜 alarm과 date를 테스트하기 위한 프로그램 alarmtest, datetest를 구현한 소스 파일

##### 2-1-1-1. int main(int argc, char \*argv[])

alarmtest.c는 과제 개요에 있는 코드를 바탕으로 그대로 작성되었다. datetest.c의 경우에는 date 시스템 콜을 호출하여 printf로 현재 시간을 출력한다.

#### 2-1-2. sys\_proc.c

추가한 시스템 콜의 동작 부분을 작성한 소스 파일

##### 2-1-2-1. int sys\_date(void)

한 개의 포인터 인자를 받는다. 자료형은 struct rtcdate이고, 커널의 cmostime 함수를 호출하여 현재 시간을 넣고 0을 반환한다.

##### 2-1-2-2. int sys\_alarm(void)

한 개의 정수 인자를 받는다. 현재 프로세스의 alarmticks를 0으로 초기화하고, alarm\_timer를 인자 \* 100으로 정한 후, 0을 반환한다.

#### 2-1-3. trap.c

alarm의 tick을 세기 위해 수정하려는 trap 함수가 있는 소스 파일

##### 2-1-3-1. void trap(struct trapframe \*tf)

가장 마지막 분기문에서 현재 프로세스가 NULL이 아니면서, 유저 모드라면 tick count를 하나 올린다. tick count가 alarm tick과 같아지면 프로세스를 종료시키면서, 문자열(SSU\_Alarm!과 현재 시간)을 출력한다.

## 2-2. 함수 호출 그래프

### 2-2-1. alarmtest

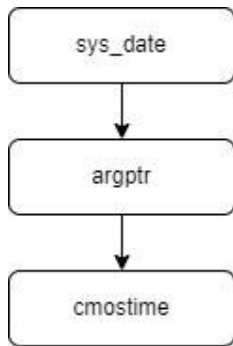
main (alarmtest.c) -> alarm (system call) -> date (system call) -> killed (exit by alarm)

### 2-2-2. datatest

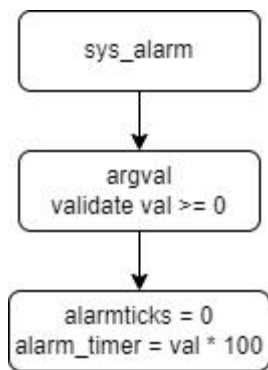
main (datatest.c) -> date (system call) -> exit

## 2-3. 순서도

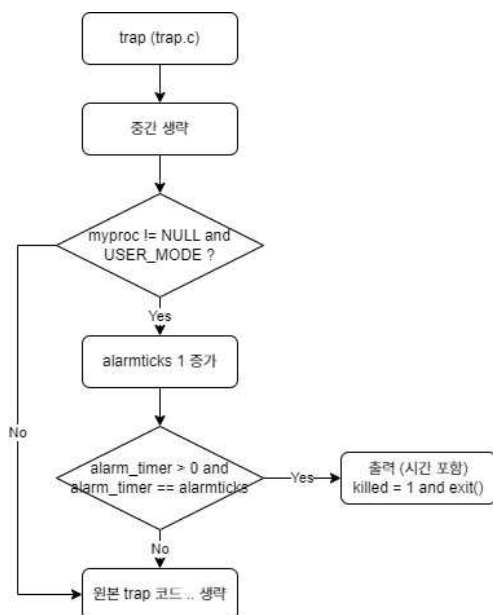
### 2-3-1. date 순서도



### 2-3-2. alarm 순서도



### 2-3-3. trap.c 순서도 (alarm tick 처리)



### 3. 실행결과

#### 3-1. datetest

```
Free: Starting sh
$ datetest
Current time : 2023-9-18 6:48:15
$
```

#### 3-2. alarmtest

##### 3-2-1. 기본 실행

```
$ alarmtest 1
SSU_Alarm Start
Current time : 2023-9-18 4:0:10
SSU_Alarm!
Current time : 2023-9-18 4:0:11
$ alarmtest 5
SSU_Alarm Start
Current time : 2023-9-18 4:0:14
SSU_Alarm!
Current time : 2023-9-18 4:0:19
$ alarmtest 10
SSU_Alarm Start
Current time : 2023-9-18 4:0:21
SSU_Alarm!
Current time : 2023-9-18 4:0:31
$
```

##### 3-2-2. [예외 처리] 인자 없거나 0 이하인 정수

```
$ alarmtest
$ alarmtest -1
SSU_Alarm Start
Current time : 2023-9-18 6:54:37

QEMU: Terminated
```

시스템 콜에서는 예외 처리하여 -1을 반환하지만,  
alarmtest.c에서 별도 처리 없이 while(1);을  
돌리기 때문에 무한 루프에 빠짐.

#### 4. 소스코드 (주석 포함)

##### 4-1. alarmtest.c

```
#include "types.h"
#include "user.h"
#include "date.h"
int main(int argc, char *argv[]) {
    int seconds;
    struct rtcdate dt;
    if (argc <= 1) exit();
    seconds = atoi(argv[1]);
    alarm(seconds);
    date(&dt);
    printf(1, "SSU_Alarm Start\n");
    printf(1, "Current time : %d-%d-%d %d:%d:%d\n", dt.year, dt.month, dt.day, dt.hour,
dt.minute, dt.second);
    while(1);
    exit();
}
```

##### 4-2. datetest.c

```
#include "types.h"
#include "user.h"
#include "date.h"
int main(int argc, char *argv[]) {
    struct rtcdate dt;
    date(&dt);
    printf(1, "Current time : %d-%d-%d %d:%d:%d\n", dt.year, dt.month, dt.day, dt.hour,
dt.minute, dt.second);
    exit();
}
```

##### 4-3. Makefile (수정한 부분만) - UPROGS

```
UPROGS=\
    _cat\

<중간 생략>

    _zombie\
    _datetest\
    _alarmtest
```

##### 4-4. proc.h (수정한 부분만) - struct proc

```
struct proc {
    uint sz;                // Size of process memory (bytes)
    pde_t* pgdir;           // Page table
    char *kstack;           // Bottom of kernel stack for this process
    enum procstate state;   // Process state
```

```

int pid;                // Process ID
struct proc *parent;    // Parent process
struct trapframe *tf;   // Trap frame for current syscall
struct context *context; // switch() here to run process
void *chan;             // If non-zero, sleeping on chan
int killed;             // If non-zero, have been killed
struct file *ofile[NOFILE]; // Open files
struct inode *cwd;      // Current directory
char name[16];          // Process name (debugging)
int alarmticks;         // Current tick
int alarm_timer;        // Current setted timer
};

```

#### 4-5. syscall.c (수정한 부분만)

##### 4-5-1. extern 함수 선언

```

extern int sys_date(void);
extern int sys_alarm(void);

```

##### 4-5-2. 테이블 목록 추가

```

[SYS_date]    sys_date,
[SYS_alarm]   sys_alarm,

```

#### 4-6. syscall.h - 시스템 콜 번호 추가

```

#define SYS_date    22
#define SYS_alarm   23

```

#### 4-7. sysproc.c - 시스템 콜 구현부

```

int
sys_date(void)
{
    struct rtcdate *dt;
    if(argptr(0, (void*)&dt, sizeof(*dt)) < 0) // validate argument
        return -1;

    cmostime(dt); // get time from cmostime
    return 0;
}

int
sys_alarm(void)
{
    int sec;
    if(argint(0, &sec) < 0)
        return -1;
    if (sec <= 0) return -1;
    myproc()->alarmticks = 0; // reset current alarm tick to 0.
    myproc()->alarm_timer = sec * 100; // 1tick = 10ms in xv6, that is 1000ms(1s, unit of
argument) equals to 100tick.
}

```

```
    return 0;
}
```

#### 4-8. trap.c

##### 4-8-1. 헤더파일 추가

```
#include "date.h"
```

##### 4-8-2. trap 함수 내부 수정

```
//PAGEBREAK: 41
void
trap(struct trapframe *tf)
{
    if(tf->trapno == T_SYSCALL){

<중간 생략>

        // Force process to give up CPU on clock tick.
        // If interrupts were on while locks held, would need to check nlock.
        if(myproc() && myproc()->state == RUNNING &&
            tf->trapno == T_IRQ0+IRQ_TIMER)
            yield();

        if (myproc() && (tf->cs & 3) == DPL_USER) {
            myproc()->alarmticks++;
            //    cprintf("%d %d\n", myproc()->alarmticks, myproc()->alarm_timer);
            if (myproc()->alarm_timer > 0 && myproc()->alarm_timer == myproc()->alarmticks) {
                cprintf("SSU_Alarm!\n"); // stdout
                struct rtcdate dt;
                cmostime(&dt); // get current time
                cprintf("Current time : %d-%d-%d %d:%d:%d\n", dt.year, dt.month, dt.day, dt.hour,
                    dt.minute, dt.second);
                myproc()->killed = 1; // set killed
                exit();
            }
        }

        // Check if the process has been killed since we yielded
        if(myproc() && myproc()->killed && (tf->cs&3) == DPL_USER)
            exit();
    }
}
```

#### 4-9. user.h - 접근용 함수 선언

```
int date(struct rtcdate*);
int alarm(int);
```

#### 4-10. usys.S - 시스템 콜 목록 추가

```
SYSCALL(date)
SYSCALL(alarm)
```