컴퓨터학부 20192403 박상철

1. 개요

시스템 호출의 동작 과정을 살펴보고 유저 영역과 커널 영역에 대해 학습하여 시스템콜 함수를 직접 추가해보는 과제

2. 상세설계

사전지식(과제 명세)

텍스트, 폰트, 화이트, 대수학이(가) 표시된 사진

자동 생성된 설명

시스템콜 함수의 동작 과정은 유저 영역에서 함수를 호출하면 커널 영역에서 시스템 콜의 과정을 처리하고 그 반환값을 다시 유저 영역으로 보내어 사용자에게 결과값을 반환한다.

2-1) datetest

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

2-2) alarm\_test

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

3. 결과

datetest

텍스트, 폰트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

alarm\_test

텍스트, 폰트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

4. 소스코드

user.h

int date(struct *rtcdate*\*);

int alarm(int);

usys.S

SYSCALL(date)

SYSCALL(alarm)

syscall.h

#define SYS\_date 22

#define SYS\_alarm 23

syscall.c

extern int sys\_date(void);

extern int sys\_alarm(void);

static int (\*syscalls[])(void) = {

...

[SYS\_date] sys\_date,

[SYS\_alarm] sys\_alarm,

};

sysproc.c

int

sys\_date(void)

{

struct *rtcdate* \*date;

if (argptr(0, (void\*)&date, sizeof(struct *rtcdate*\*)) < 0)

return -1;

cmostime(date);

return 0;

}

int sys\_alarm(void) {

int time\_interval;

if (argint(0, &time\_interval) < 0)

return -1;

// 시간(초)을 틱 수로 변환하여 알람 시간으로 설정

myproc()->alarmticks = 0;

myproc()->alarm\_timer = time\_interval\*100;

return 0;

}

proc.h

struct *proc* {

*...*

int alarmticks; // 알람까지 남은 틱 수

int alarm\_timer; // 알람이 설정된 시간(틱 수)

};

trap.c

switch(*tf*->trapno){

case T\_IRQ0 + IRQ\_TIMER:

if(cpuid() == 0){

acquire(&tickslock);

ticks++;

// 프로세스가 존재하고 사용자 모드인 경우만 알람 처리

if (myproc() && (*tf*->cs & 3)) {

if (myproc()->alarm\_timer > 0) {

myproc()->alarmticks++;

if (myproc()->alarmticks >= myproc()->alarm\_timer) {

// 알람 시간이 경과하면 현재 프로세스를 종료

myproc()->killed = 1;

struct *rtcdate* r;

cmostime(&r);

cprintf("SSU\_Alarm!\n");

cprintf("Current time : %d-%d-%d %d:%d:%d\n", r.year, r.month, r.day, r.hour, r.minute, r.second);

}

}

}

wakeup(&ticks);

release(&tickslock);

}

lapiceoi();

break;

datetest.c

#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();

}

printf(1, "Current time : %d-%d-%d %d:%d:%d\n", r.year,r.month,r.day,r.hour,r.minute,r.second);

exit();

}

alarm\_test.c

#include "types.h"

#include "user.h"

#include "date.h"

int main(int *argc*, char \**argv*[])

{

int seconds;

struct *rtcdate* r;

if(*argc* <= 1)

exit();

seconds = atoi(*argv*[1]);

alarm(seconds);

date(&r);

printf(1, "SSU\_Alarm Start\n");

printf(1, "Current time : %d-%d-%d %d:%d:%d\n",r.year,r.month,r.day,r.hour,r.minute,r.second);

while (1)

;

exit();

}

makefile

UPROGS=\

\_cat\

\_echo\

\_forktest\

\_grep\

\_init\

\_kill\

\_ln\

\_ls\

\_mkdir\

\_rm\

\_sh\

\_stressfs\

\_usertests\

\_wc\

\_zombie\

\_helloxv6\

\_htac\

\_datetest\

\_alarm\_test\