컴퓨터학부 20152385 송민구

1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/time.h>

#include <errno.h>

#include <setjmp.h>

#include <signal.h>

#include <time.h>

#include "ssu\_runtime.h"

static void ssu\_alarm(int signo);

static void ssu\_func(int signo);

void ssu\_mask(const char \*str);

static volatile sig\_atomic\_t can\_jump;

static sigjmp\_buf jump\_buf;

int main(void)

{

gettimeofday(&begin\_t, NULL);

if (signal(SIGUSR1, ssu\_func) == SIG\_ERR) { // SIGUSR1의 동작을 설정

fprintf(stderr, "SIGUSR1 error\n");

exit(1);

}

if (signal(SIGALRM, ssu\_alarm) == SIG\_ERR) { // SIGALRM의 동작을 설정

fprintf(stderr, "SIGALRM error\n");

exit(1);

}

ssu\_mask("starting main: ");

if (sigsetjmp(jump\_buf, 1)) { // sigsetjmp()으로 프로세스의 현재상태 저장

ssu\_mask("ending main: ");

gettimeofday(&end\_t, NULL);

ssu\_runtime(&begin\_t, &end\_t);

exit(0);

}

can\_jump = 1;

while (1)

pause(); // signal을 기다림

gettimeofday(&end\_t, NULL);

ssu\_runtime(&begin\_t, &end\_t);

exit(0);

}

void ssu\_mask(const char \*str)

{

sigset\_t sig\_set;

int err\_num;

err\_num = errno; // 에러넘버가 바뀔 수 있으니에러넘버를 저장

if (sigprocmask(0, NULL, &sig\_set) < 0) {

fprintf(stderr, "sigprockmask() error\n");

exit(1);

}

printf("%s", str);

if (sigismember(&sig\_set, SIGINT)) {

printf("SIGINT ");

}

if (sigismember(&sig\_set, SIGQUIT)) {

printf("SIGQUIT ");

}

if (sigismember(&sig\_set, SIGUSR1)) {

printf("SIGUSR1 ");

}

if (sigismember(&sig\_set, SIGALRM)) {

printf("SIGALRM ");

}

printf("\n");

errno = err\_num; // 원래 에러넘버로 재지정

}

static void ssu\_func(int signo)

{

time\_t start\_time;

if (can\_jump == 0) { // can\_jump변수가 0이라면 그냥 리턴

return;

}

ssu\_mask("starting ssu\_func: ");

alarm(3);

start\_time = time(NULL);

while (1) {

if (time(NULL) > start\_time + 5) {

break;

}

}

ssu\_mask("ending ssu\_func: ");

can\_jump = 0; // can\_jump변수를 0으로

siglongjmp(jump\_buf, 1); // 저장해둔곳으로 이동

}

static void ssu\_alarm(int signo)

{

ssu\_mask("in ssu\_alarm: ");

}

1. 실행 결과

