컴퓨터학부 20152385 송민구

1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/time.h>

#include <signal.h>

#include "ssu\_runtime.h"

void ssu\_check\_pending(int signo, char \*signame);

void ssu\_signal\_handler(int signo);

int main(void)

{

struct sigaction sig\_act;

sigset\_t sig\_set;

gettimeofday(&begin\_t, NULL);

sigemptyset(&sig\_act.sa\_mask);

sig\_act.sa\_flags = 0;

sig\_act.sa\_handler = ssu\_signal\_handler;

if (sigaction(SIGUSR1, &sig\_act, NULL) != 0) { // SIGUSR1의 동작 설정

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

exit(1);

}

else {

sigemptyset(&sig\_set);

sigaddset(&sig\_set, SIGUSR1);

if (sigprocmask(SIG\_SETMASK, &sig\_set, NULL) != 0) { // SIGUSR1을 블록

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

exit(1);

}

else {

printf("SIGUSR1 signals are now blocked\n");

kill(getpid(), SIGUSR1);

printf("after kill()\n");

ssu\_check\_pending(SIGUSR1, "SIGUSR1");

sigemptyset(&sig\_set);

sigprocmask(SIG\_SETMASK, &sig\_set, NULL); // 블록해제

printf("SIGUSR1 signals are no longer blocked\n");

ssu\_check\_pending(SIGUSR1, "SIGUSR1");

}

}

gettimeofday(&end\_t, NULL);

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

exit(0);

}

void ssu\_check\_pending(int signo, char \*signame) {

sigset\_t sig\_set;

if (sigpending(&sig\_set) != 0) { // sigpending()으로 펜딩중인지 검사

printf("sigpending() error\n");

}

else if (sigismember(&sig\_set, signo)) {

printf("a %s signal is pending\n", signame);

}

else {

printf("%s signals are not pending\n", signame);

}

}

void ssu\_signal\_handler(int signo) {

printf("in ssu\_signal\_handler function\n");

}

1. 실행 결과

