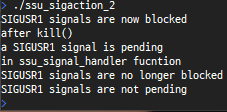
컴퓨터학부 20162448 김병준

1. 결과



1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <signal.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;  sigemptyset(&sig\_act.sa\_mask); // 시그널 집합 초기화  sig\_act.sa\_flags = 0;  sig\_act.sa\_handler = ssu\_signal\_handler;  if (sigaction(SIGUSR1, &sig\_act, NULL) != 0) { // SIGUSR1 시그널 처리 등록  fprintf(stderr, "sigaction() error\n");  exit(1);  }  else {  sigemptyset(&sig\_set); // 시그널 집합 초기화  sigaddset(&sig\_set, SIGUSR1); // 시그널 등록  if (sigprocmask(SIG\_SETMASK, &sig\_set, NULL) != 0) { // sig\_set procmask..  fprintf(stderr, "sigprocmask() error\n");  exit(1);  }  else {  printf("SIGUSR1 signals are now blocked\n");  kill(getpid(), SIGUSR1); // SIGUSR1 시그널 전달  printf("after kill()\n");  ssu\_check\_pending(SIGUSR1, "SIGUSR1"); // SIGUSR1 시그널 유무 확인  sigemptyset(&sig\_set); // 시그널 집합 초기화  sigprocmask(SIG\_SETMASK, &sig\_set, NULL); // 블록 시그널 설정  printf("SIGUSR1 signals are no longer blocked\n");  ssu\_check\_pending(SIGUSR1, "SIGUSR1"); // SIGUSR1 시그널 유무 확인  }  }  exit(0);  }  void ssu\_check\_pending(int signo, char \*signame) {  sigset\_t sig\_set;  if (sigpending(&sig\_set) != 0)  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 fucntion\n");  } |