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;  //sig\_act 구조체를 설정  sigemptyset(&sig\_act.sa\_mask);  sig\_act.sa\_flags = 0;  sig\_act.sa\_handler = ssu\_signal\_handler;  //SIGUSR1 시그널이 발생했을 때의 엑션에 대한 정보 전달  if(sigaction(SIGUSR1, &sig\_act, NULL) != 0) {  fprintf(stderr, "sigaction() error\n");  exit(1);  }  else {  //sig\_set을 0으로 초기화  sigemptyset(&sig\_set);  //sig\_set에 SIGUSR1 시그널 추가  sigaddset(&sig\_set, SIGUSR1);  //sig\_set에 있는 시그널을 마스크함  if(sigprocmask(SIG\_SETMASK, &sig\_set, NULL) != 0) {  fprintf(stderr, "sigprocmask() error\n");  exit(1);  }  else {  printf("SIGUSR1 signals are now blocked\n");  kill(getpid(), SIGUSR1);//시그널이 마스크됨  printf("after kill()\n");  ssu\_check\_pending(SIGUSR1, "SIGUSR1");//유보된 시그널이 존재  sigemptyset(&sig\_set);//sig\_set을 0으로 초기화  //sig\_set에 시그널이 없어 마스크되는 시그널이 없음  sigprocmask(SIG\_SETMASK, &sig\_set, NULL);  printf("SIGUSR1 signals are no longer blocked\n");  ssu\_check\_pending(SIGUSR1, "SIGUSR1");//유보된 시그널이 없음  }  }  exit(0);  }  void ssu\_check\_pending(int signo, char \*signame) {  sigset\_t sig\_set;  //유보 중인 시그널이 있는지 확인  if(sigpending(&sig\_set) != 0)  printf("sigpending() error\n");  else if(sigismember(&sig\_set, signo))//유보된 시그널이 있다면  printf("a %s signal is pending\n", signame);  else//유보된 시그널이 없다면  printf("%s signals are not pending\n", signame);  }  void ssu\_signal\_handler(int signo) {  printf("in ssu\_signal\_handler function\n");  } |

2. 실행결과

