1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <signal.h>  void ssu\_signal\_handler(int signo) {  printf("ssu\_signal\_handler control\n");  }  int main(void) {  struct sigaction sig\_act;  sigset\_t sig\_set;  //sig\_act.sa\_mask를 0으로 초기화  sigemptyset(&sig\_act.sa\_mask);  //sig\_act 구조체 값 설정  sig\_act.sa\_flags = 0;  sig\_act.sa\_handler = ssu\_signal\_handler;  //SIGUSR1 시그널이 발생했을 때의 엑션에 대한 정보 전달  sigaction(SIGUSR1, &sig\_act, NULL);  printf("before first kill()\n");  //현재 프로세스에 SIGUSR1 시그널을 발생시킴  kill(getpid(), SIGUSR1);  //sig\_set을 0으로 초기화  sigemptyset(&sig\_set);  //SIGUSR1 시그널을 sig\_set에 추가  sigaddset(&sig\_set, SIGUSR1);  //sig\_set에 있는 시그널을 마스크함  sigprocmask(SIG\_SETMASK, &sig\_set, NULL);  printf("before second kill()\n");  kill(getpid(), SIGUSR1);//마스크가 되어 시그널 핸들러가 수행되지 않음  printf("after second kill()\n");  exit(0);  } |

2. 실행결과

