1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <signal.h>  #include <sys/types.h>  #include <sys/wait.h>  static void ssu\_signal\_handler1(int signo);  static void ssu\_signal\_handler2(int signo);  int main(void) {  struct sigaction act\_int, act\_quit;  //act\_int의 시그널 핸들러 등록  act\_int.sa\_handler = ssu\_signal\_handler1;  //act\_int.sa\_mask를 0으로 초기화  sigemptyset(&act\_int.sa\_mask);  //act\_int.sa\_mask에 SIGQUIT 시그널을 추가  sigaddset(&act\_int.sa\_mask, SIGQUIT);  act\_int.sa\_flags = 0;  //SIGINT 시그널이 발생했을 때의 엑션에 대한 정보 전달  if(sigaction(SIGINT, &act\_int, NULL) < 0) {  fprintf(stderr, "sigaction(SIGINT) error\n");  exit(1);  }  //act\_quit의 시그널 핸들러 등록  act\_quit.sa\_handler = ssu\_signal\_handler2;  //act\_quit.sa\_mask를 0으로 초기화  sigemptyset(&act\_quit.sa\_mask);  //act\_quit.sa\_mask에 SIGINT 시그널을 추가  sigaddset(&act\_quit.sa\_mask, SIGINT);  act\_quit.sa\_flags = 0;  //SIGQUIT 시그널이 발생했을 때의 엑션에 대한 정보 전달  if(sigaction(SIGQUIT, &act\_quit, NULL) < 0) {  fprintf(stderr, "sigaction(SIGQUIT) error\n");  exit(1);  }  pause();  exit(0);  }  static void ssu\_signal\_handler1(int signo) {  printf("Signal handler of SIGINT : %d\n", signo);  printf("SIGQUIT signal is blocked : %d\n", signo);  printf("sleeping 3 sec\n");  sleep(3);  printf("Signal handler of SIGINT ended\n");  }  static void ssu\_signal\_handler2(int signo) {  printf("Signal handler of SIGQUIT : %d\n", signo);  printf("SIGINT signal is blocked : %d\n", signo);  printf("sleeping 3 sec\n");  sleep(3);  printf("Signal handler of SIGQUIT ended\n");  } |

2. 실행결과

