1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <setjmp.h>  #include <signal.h>  static void ssu\_signal\_handler1(int signo);  static void ssu\_signal\_handler2(int signo);  sigjmp\_buf jmp\_buf1;  sigjmp\_buf jmp\_buf2;  int main(void) {  struct sigaction act\_sig1;  struct sigaction act\_sig2;  int i, ret;  printf("My PID is %d\n", getpid());  ret = sigsetjmp(jmp\_buf1, 1);  //sigsetjmp() 직접 호출 시  if(ret == 0) {  act\_sig1.sa\_handler = ssu\_signal\_handler1;  //SIGINT 시그널이 발생했을 때의 엑션을 설정  sigaction(SIGINT, &act\_sig1, NULL);  }  else if(ret == 3)//siglongjmp()에서 호출 시  printf("-------------\n");    printf("Starting\n");  sigsetjmp(jmp\_buf2, 2);  act\_sig2.sa\_handler = ssu\_signal\_handler2;  //SIGUSR1 시그널이 발생했을 때의 엑션을 설정  sigaction(SIGUSR1, &act\_sig2, NULL);  for(i = 0; i < 20; i++) {  printf("i = %d\n", i);  sleep(1);  }    exit(0);  }  //SIGINT 시그널의 시그널 핸들러 함수  static void ssu\_signal\_handler1(int signo) {  fprintf(stderr, "\nInterrupted\n");  //sigsetjmp() 호출, 반환값은 3  siglongjmp(jmp\_buf1, 3);  }  //SIGUSR1 시그널의 시그널 핸들러 함수  static void ssu\_signal\_handler2(int signo) {  fprintf(stderr, "\nSIGUSR1\n");  //sigsetjmp() 호출, 반환값은 2  siglongjmp(jmp\_buf2, 2);  } |

2. 실행결과

