1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <setjmp.h>  void ssu\_nested\_func(int loc\_var, int loc\_volatile, int loc\_register);  static jmp\_buf glob\_buffer;  int main(void) {  //최적화 시 레지스터에 저장  register int loc\_register;  volatile int loc\_volatile;  //최적화 시 레지스터에 저장  int loc\_var;  loc\_var = 10;  loc\_volatile = 11;  loc\_register = 12;  //longjmp()에 의해 호출되었을 경우  if(setjmp(glob\_buffer) != 0) {  printf("after longjmp, loc\_var = %d, loc\_volatile = %d, loc\_register = %d\n", loc\_var, loc\_volatile, loc\_register);  exit(0);  }  loc\_var = 80;  loc\_volatile = 81;  loc\_register = 83;  ssu\_nested\_func(loc\_var, loc\_volatile, loc\_register);  exit(0);  }  void ssu\_nested\_func(int loc\_var, int loc\_volatile, int loc\_register) {  printf("before longjmp, loc\_var = %d, loc\_volatile = %d, loc\_register = %d\n", loc\_var, loc\_volatile, loc\_register);  //setjmp()로 이동, 호출된 setjmp()는 1을 반환  longjmp(glob\_buffer, 1);  } |

2. 실행결과

컴파일 시 최적화 옵션을 설정하지 않으면 지역변수, volatile변수, register변수가 모두 메모리에 저장되어 longjmp() 호출 시의 값을 유지

컴파일 시 최적화 옵션을 설정하면 지역변수, register변수는 레지스터에 저장되어 setjmp() 호출 시의 값으로 돌아가고 volatile변수는 그대로 메모리에 저장되어 longjmp() 호출 시의 값을 그대로 유지

