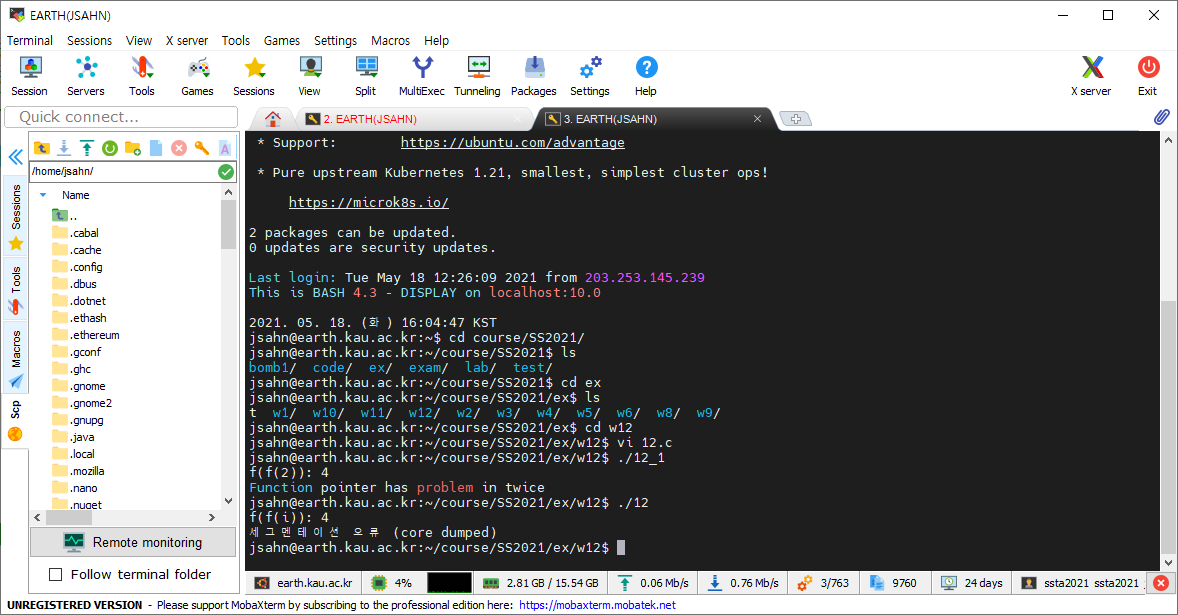
**고급시스템프로그래밍 12주차 실시간수업 과제**

|  |
| --- |
| 제출일: 2021.05.27  이름: 김현용 |

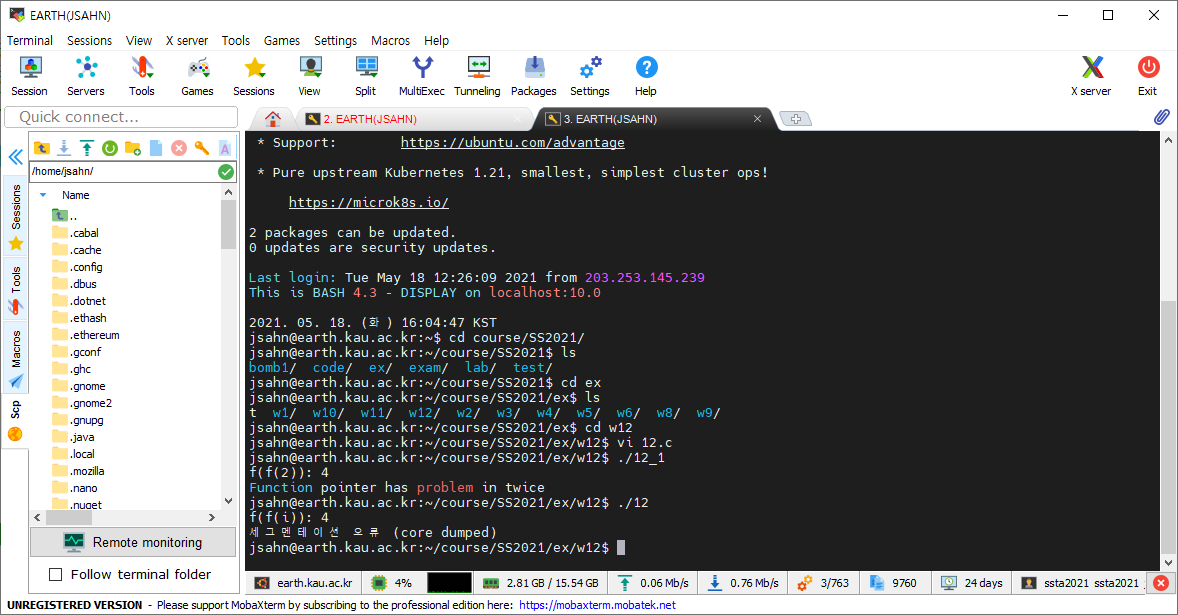
1. 다음 프로그램의 twice 함수는 함수 인자를 받아 정수 인자에 두 번 적용한 결과를 출력한다. Main 함수의 두 번째 twice 함수 호출 시 잘못된 함수 주소를 전달하여 segmentation fault가 발생한다. 이에 대하여 아래와 같이 twice 수행 중에 segmentation fault 발생하면 다음과 같은 특별한 메시지를 출력하고 프로그램을 종료하도록 signal을 사용해 프로그램을 수정하시오. (main 함수는 고치지 않는다.) 단 다른 종류의 segmentation fault에 대한 출력은 바뀌면 안된다.

|  |
| --- |
| #include <stdio.h>  void twice(int (\*f)(int), int i) {  printf("f(f(i)): %d\n", f(f(i)));  }  int addone(int i) {  return i+1;  }  int main() {  int (\*f)(int);  int \*i;  f = addone;  twice(f, 2);  f = 0;  twice(f, 2);  f = addone;  twice(f, 3);  i = 0;  f = addone;  twice(f, \*i);  return 0;  } |

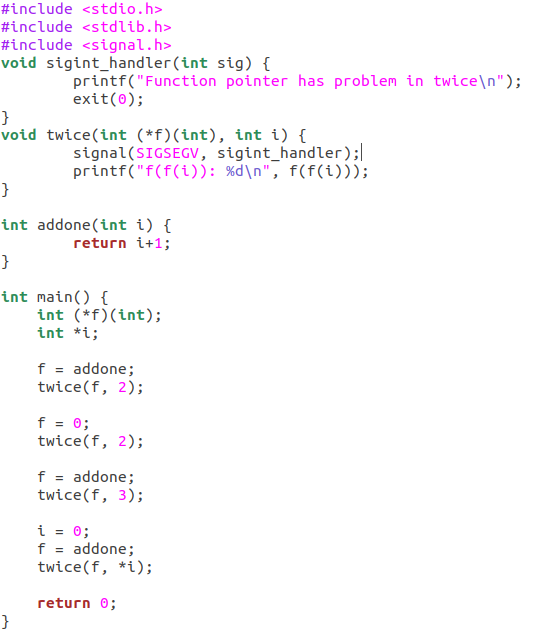
(기존 실행화면)



(수정 후 실행화면)



A1-1) 코드

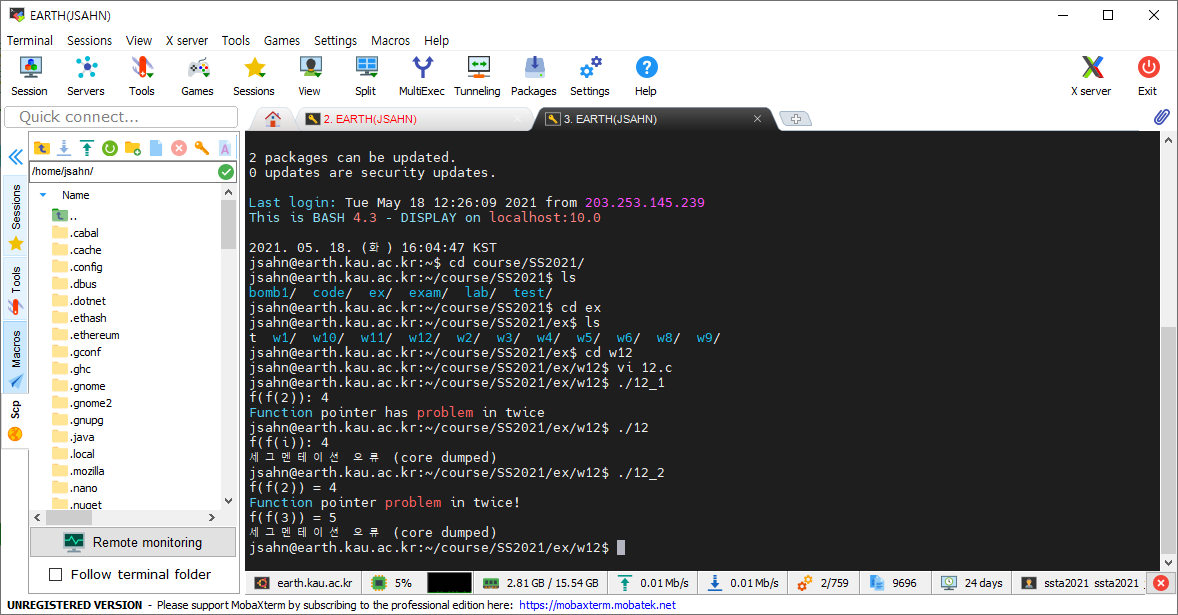


A1-2) 결과 화면

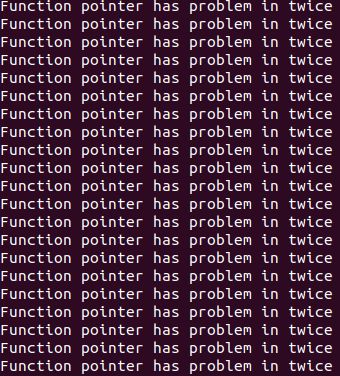


A1-3) 코드 설명  
코드를 보면 f = addone함수의 시작 주소로 하여 twice함수를 호출하면 원래는 printf문으로 f(f(i))를 계산하여 출력하는 것인데, 이 후, f = 0으로 바꾸기 때문에, 0은 함수의 시작 주소가 아니기 때문에, Segmentation fault가 발생하게 되는 것이다. 이 때, Segmentation fault 오류 문을 출력하는 것은 해당하는 Signal을 보내기 때문인데, 그에 해당하는 시그널인 SIGSEGV의 signal handler를 위의 선언한 sigint\_handler로 바꾸어 주어 원하는 출력문을 printf하고 프로세스를 종료하는 exit(0)을 수행하여 종료한다.

1. 위 1번의 프로그램에 대하여 twice 수행 중 segmentation fault가 발생하면 해당 twice 함수는 그냥 반환되고 다음 명령어가 계속 수행되도록 하려 한다.

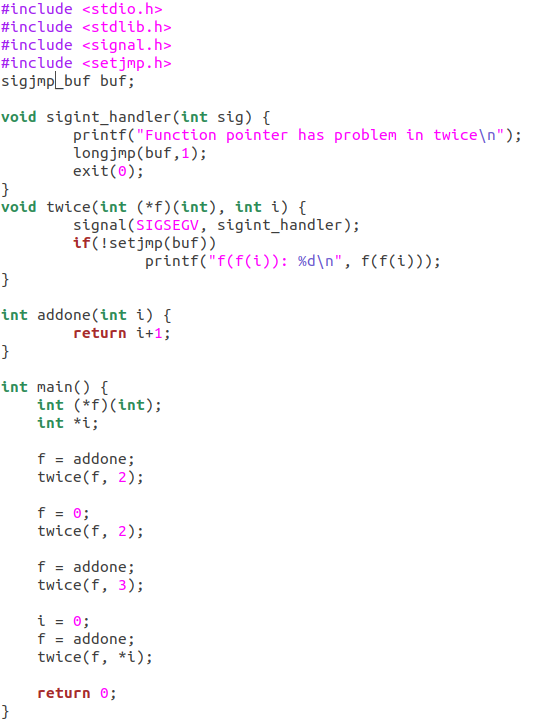


1. 만약 signal handler에서 프로그램을 종료하지 않고 리턴하면 어떻게 되는가?

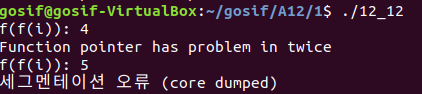
  
sigint\_handler의 exit(0)을 지울 경우 핸들러가 종료되면 그 오류가 일어나기 이전으로 돌아가기 때문에 (printf문) 매번 printf의 f를 호출하면서 Segmentation fault가 발생해서 같은 오류문이 계속 출력된다.

1. 1)과 같이 잘못된 함수 호출이 계속 되지 않도록 함수 호출 전의 과거로 돌아가 함수 호출을 건너뛰도록 하여야 한다. 이를 위하여 setjmp와 longjmp를 사용하여 프로그램을 수정하시오.

A2-1) 코드



A2-2) 결과 화면



A2-3) 코드 설명

처음부터 보면 main에서 f = addone; twice(f, 2)는 문제 없으며 출력 또한 4로 잘 출력되었다. 그 다음 줄은 f = 0으로 바꾸었기 때문에 twice 함수 안의 printf문에서 f를 호출하면서 segmentation fault가 나서 handler로 가 오류문을 출력했다. 그리고 그 다음도 twice(f,3)도 5로 잘 출력이 되었고, 그 다음 줄에서는 i는 선언할 때 int 형이었는데, 입력으로 int \* type을 입력했기 때문에 main함수에서 segmentation fault가 발생해서 원래의 세그멘테이션 오류를 출력하게 되었다.