Что на самом деле содержится в jmp_buf при использовании setjmp и longjmp?

setjmp () должен сохранять регистры, включая «адрес возврата» и «указатель стека», в «jmp_buf». Когда я компилирую (как gcc, так и clang) и отлаживаю следующую программу под x86_64 с помощью glibc, я не могу понять, что находится в «jmp_buf» и где «адрес возврата» и «указатель стека» расположены в «jmp_buf».

```
#include <stdio.h>
#include <setjmp.h>

int main()
{
    int i;

    jmp_buf env;

    i = setjmp(env);

    printf("i = %d\n", i);

    if (i != 0) return;

    longjmp(env, 2);
    printf("Does this line get printed?\n");
}
```

Когда программа останавливается в точке останова перед printf ("i =% d \ n", i); ", я пробовал функциональность gdb:" p / x env "; однако я не могу найти «RIP возврата» и «предыдущий RSP» в этой структуре (env), которая содержит __jmpbuf и __saved_mask. Кто-нибудь знает, как именно работают эти две функции и что именно они сохраняют под x86_64 с помощью glibc (я использую ubuntu 14.04)?

c debugging gdb setjmp

спросил 21 '15 янв в 3:14

WindChaser
8901927

1

Я видел несколько подобных вопросов на stackoverflow.com, но ответы обычно такие: «Это зависит от архитектуры, ОС, библиотеки и ABI ...» или вставьте стандарт. Это не то, что я хотел бы знать. – WindChaser 21 янв.

Почему бы вам не взглянуть на реализацию на своей платформе? Это открытый исходный код. Например: github.com/lattera/glibc/blob/master/sysdeps/sh/… – это то, что longjmp вызывает в glibc. – John Zwinck 21 янв.

Верьте или нет, содержание `jmp_buf` является преднамеренно бессмысленным. Если вы посмотрите на исходный код x86_64 `setjmp()` , вы заметите несколько ссылок на `PTR_MANGLE` . Это внутренний макрос glibc, который выполняет XOR для локального значения потока по отношению к регистру. Это используется здесь в значительной степени для того, чтобы разработчики не полагались на макет `jmpbuf` - это считается деталью реализации и может меняться в зависимости от версии libc.

Если вы хотите что-то читабельное, обратите внимание на интерфейс ucontext .

ответил 21 '15 янв в 3:32
user149341

2

Я считаю, что изменение указателя было введено как функция безопасности, а не для того, чтобы скрыть структуру jmpbuf. – Employed Russian 21 янв.

@EmployedRussian Вы имеете в виду, что это может предотвратить некоторые атаки, такие как переполнение стека? – WindChaser 22 янв.

@WindChaser Это единственная соответствующая ссылка, которую я смог найти: udrepper.livejournal.com/13393.html – Employed Russian 22 янв.

Соответствующая ветка списка рассылки: redhat.com/archives/rhl-devel-list/2006-September/msg00499.html – user149341 22 янв.

Связанный

2728 В чем разница между #include <filename> и #include "filename"?

1016 В чем разница между ++ i и i ++?

867 Почему эти конструкции используют неопределенное поведение до и после инкремента?

инкремента?

1582 В чем разница между const int *, const int * const и int const *?

958 В чем разница между определением и декларацией?

4 pthreads, setjmp, longjmp. Как узнать, когда функция завершена?

8 Предупреждение "может быть затерто" на объекте C ++ с помощью setjmp

116 Практическое использование setjmp и longjmp в C

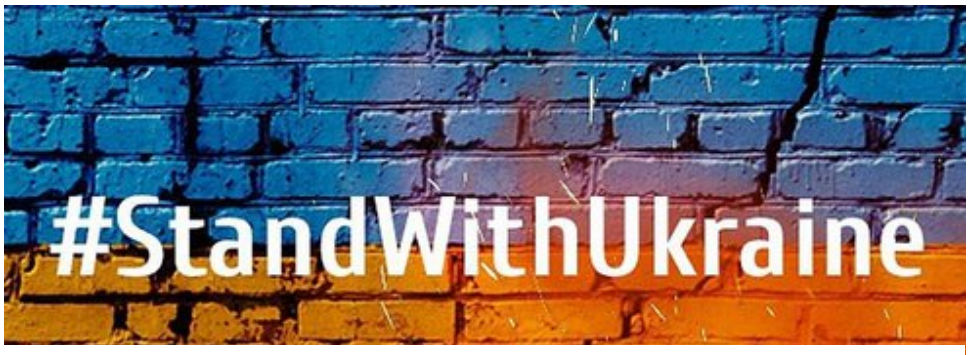
0 ошибка сегментации пользовательской реализации setjmp / longjmp

Ваш Ваненька? У нас его паспорт

ОБЩЕСТВО

Устал и прилег отдохнуть: где найти свой груз 200

ОБЩЕСТВО



Правда про войну на Украине

ОБЩЕСТВО

Здесь можно опознать русских пленных и 200-х

ОБЩЕСТВО

Русские люди, не молчите!

ОБЩЕСТВО

Полный список 200-х: смотри фото

ОБЩЕСТВО

Разве Россия посылала в Украину Саб-Зиро?

ОБЩЕСТВО

Груз 200: ищи своих по фото

ОБЩЕСТВО

Такой судьбы россияне хотят своим детям?

ОБЩЕСТВО

Прикройте тылы, проверьте доставку своего груза 200

ОБЩЕСТВО