



Setjmp-inspired callbacks in C

Mark Mendell
Handmade NYC Meetup 6/29/2023



setjmp + longjmp = super goto

```
jmp_buf buf;
```

```
void  
bar(void)  
{  
    puts("bar");  
    longjmp(buf, 1)  
}
```

```
void  
foo(void)  
{  
    puts("foo");  
    bar();  
}
```

```
int  
main(void)  
{  
    int ret = setjmp(buf);  
    printf("setjmp: %d\n", ret);  
    if (!ret) foo();  
    puts("done");  
}
```

Output:

```
setjmp: 0  
foo  
bar  
setjmp: 1  
done
```

Callback comparisons

Javascript

```
let button = createElement("Click me");
parent.appendChild(button);
button.onclick = () => {
    console.log("clicked");
    doStuff();
};
```

Simple C

```
double x0button, y0button, x1button, y1button;

void
draw(void)
{
    drawbutton(x, y, "Click me", &x0button, &y0button, &x1button, &y1button);
}

void
onclick(double x, double y)
{
    if (x>x0button && x<x1button && y>y0button && y<y1button) {
        puts("clicked");
        dostuff();
    }
}
```

Dear ImGui

```
void
draw(void)
{
    if (ImGui::Button("Click me")) {
        puts("clicked");
        dostuff();
    }
}
```

Setjmp

```
void
draw(void)
{
    if (setjmpclickdrawtext(x,y,"Click me")) {
        puts("clicked");
        dostuff();
        longjmpppop();
    }
}
```



Dear ImGui vs Setjmp

Dear ImGui

Frame 1: Draw() is called, button is drawn

Click: Click coordinates are saved

Frame 2: Draw() is called, runs click logic

Setjmp

Frame 1: Draw() is called, button is drawn

Click: Runs click logic



How it works

Source code:

```
void
draw(void)
{
    if (setjmpclick(10,10,20,20)) {
        puts("clicked");
        dostuff();
        longjmppop();
    }
}

void
onclick(double x, double y)
{
    for click: if x,y in click.x0,y0,x1,y1 {
        if (!setjmppush()) longjmp(click.location);
    }
}
```

Execution:

```
void
draw(void)
{
    if (setjmpclick(10,10,20,20)) {
        callback:
        puts("clicked");
        dostuff();
        longjmppop();
        goto aftercallback;
    }
}

void
onclick(double x, double y)
{
    for click: if x,y in click.x0,y0,x1,y1 {
        goto callback;
        aftercallback::
    }
}
```



Can't actually use setjmp + longjmp for this

```
jmp_buf buf;
```

Callstack:

```
void  
bar(void)  
{  
    puts("bar");  
    longjmp(buf, 1)  
}
```

```
void  
foo(void)  
{  
    puts("foo");  
    bar();  
}
```

– main ← **STACK HERE**

<main data>

<main's return address>

```
int  
main(void)  
{  
    int ret = setjmp(buf);  
    printf("setjmp: %d\n", ret);  
    if (!ret) foo();  
    puts("done");  
}
```

Can't actually use setjmp + longjmp for this

```
jmp_buf buf;
```

Callstack:

```
void  
bar(void)  
{  
    puts("bar");  
    longjmp(buf, 1)  
}
```

– foo ← STACK HERE

<foo data>

<foo's return address>

– main

<main data>

<main's return address>

```
void  
foo(void)  
{  
    puts("foo");  
    bar();  
}  
  
int  
main(void)  
{  
    int ret = setjmp(buf);  
    printf("setjmp: %d\n", ret);  
    if (!ret) foo();  
    puts("done");  
}
```

Can't actually use setjmp + longjmp for this

```
jmp_buf buf;
```

```
void  
bar(void)  
{  
    puts("bar");  
    longjmp(buf, 1)  
}
```

```
void  
foo(void)  
{  
    puts("foo");  
    bar();  
}
```

```
int  
main(void)  
{  
    int ret = setjmp(buf);  
    printf("setjmp: %d\n", ret);  
    if (!ret) foo();  
    puts("done");  
}
```

Callstack at longjmp time:

- bar ← **STACK HERE**
- <bar data>
- <bar's return address>
- foo
- <foo data>
- <foo's return address>
- main
- <main data>
- <main's return address>

Callstack after longjmp

- ~~bar~~
- <~~bar data~~>
- <~~bar's return address~~>
- ~~foo~~
- <~~foo data~~>
- <~~foo's return address~~>
- main ← **STACK HERE**
- <main data>
- <main's return address>

Setjmp simply saves and restores the stack pointer

Can't actually use setjmp + longjmp for this

```
void
draw(void)
{
    if (setjmpclick(10,10,20,20)) {
        puts("clicked");
        dostuff();
        longjmppop();
    }
}
```

Callstack at longjmp time:

– onclick ← **STACK HERE**
<onclick data>
<onclick's return address>

Callstack after longjmp

????? ← **STACK HERE**

```
void
onclick(double x, double y)
{
    for click: if x,y in click.x0,y0,x1,y1 {
        if (!setjmppush()) longjmp(click.location);
    }
}
```

Callstack at longcall time:

– onclick ← **STACK HERE**
<onclick data>
<onclick's return address>

Callstack after longcall

– draw ← **STACK HERE**
<draw data>
<draw's return address>
– onclick
<onclick's data>
<onclick's return address>

Callstack after longjmppop

~~– draw~~
~~<draw data>~~
~~<draw's return address>~~
– onclick ← **STACK HERE**
<onclick data>
<onclick's return address>

```
void
onclick(double x, double y)
{
    for click: if x,y in click.x0,y0,x1,y1 {
        if (!setjmppush()) longcall(click.location);
    }
}
```