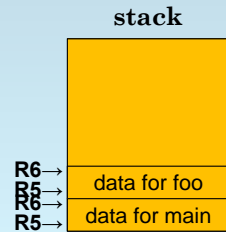


## When `main` Calls `foo`, `foo`'s Stack Frame is Pushed

When a **C** program starts, the function `main` is executed (`main`'s frame pushed on stack).

`main` may call another function, such as `foo`.



## Each Function Call Pushes Another Stack Frame

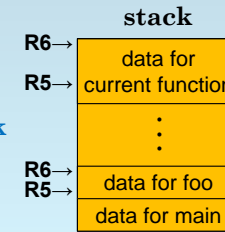
When a **C** program starts, the function `main` is executed (`main`'s frame pushed on stack).

`main` may call another function, such as `foo`.

... which calls another ...

**R5 always points to the current function's stack frame.**

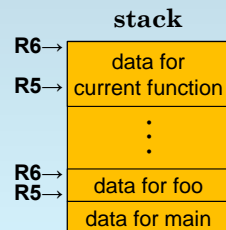
(How, exactly? Later.)



## A Function's Stack Frame is Popped When It Returns

When a function finishes executing, its stack frame is removed from the stack.

Here, execution has returned to the function `foo` called from `main`.



## Example: One Stack Frame per Function Called

Let's do an example.

```
int main ()
{
    int32_t a = 42;
    printf ("%d", a);
    return 0;
}
```

`main`'s stack frame is first.

