

Assembly in macOS: Hello World

This as code creates a “Hello, World!” program. It uses system calls to write the message to the console and then exit. The `.text` section holds the executable instructions, starting at `_start`. It first sets up the `sys_write` call (printing to `stdout`) and then the `sys_exit` call (exiting cleanly). The `.data` section stores the message itself and calculates its length.

```
.global _start
.text

_start:
    // Write "Hello, World!" to stdout
    mov x8, #64           // syscall number for write (sys_write)
    mov x0, #1            // file descriptor 1 (stdout)
    ldr x1, =msg          // address of the message
    mov x2, #14           // length of the message
    svc #0               // invoke syscall

    // Exit the program
    mov x8, #93          // syscall number for exit (sys_exit)
    mov x0, #0           // exit code 0
    svc #0              // invoke syscall

.data
msg:
    .asciz "Hello, World!\n"

as -o hello.o hello.s

% clang -o hello hello.o -nostdlib -e _start -Wl,-platform_version,macos,15.0,15.0 -arch arm64

% ./hello
zsh: invalid system call ./hello
```