

## Objective

- Investigate how the ARM Procedure Call Standard maps to actual code generation requirements

## Deep Magic

Section 5.4 of the Procedure Call Standard says:

“A Composite Type larger than 4 bytes, or whose size cannot be determined statically by both caller and callee, is stored in memory at an address passed as an extra argument when the function was called.”

1. Type in the following code:

```
typedef struct {
    int x;
    int y;
    int z;
} Point3D;

static Point3D currPoint = { 0, 0, 0 };

Point3D moveXYZ(int dx, int dy, int dz)
{
    currPoint.x += dx;
    currPoint.y += dy;
    currPoint.z += dz;

    return currPoint;
}
```

Compile the code and study the assembly-language output:

```
arm-elf-gcc -O2 -c -Wa,-a=test.lst test.c
```

2. Which registers had to be saved/restored on the stack in this function?
3. Which registers are used to pass the 3 parameters to this function?
4. How is the return value of the function (an entire structure is returned) passed back to the caller?