常见指令

- **mov**： 将某一寄存器的值复制到另一寄存器（只能用于寄存器与寄存器或者寄存器与常量之间传值，不能用于内存地址），如：

```
mov x1, x0          将寄存器 x0 的值复制到寄存器 x1 中
```

- **add**： 将某一寄存器的值和另一寄存器的值 相加 并将结果保存在另一寄存器中，如：

```
add x0, x1, x2      将寄存器 x1 和 x2 的值相加后保存到寄存器 x0 中
```

- **sub**： 将某一寄存器的值和另一寄存器的值 相减 并将结果保存在另一寄存器中：

```
sub x0, x1, x2          将寄存器 x1 和 x2 的值相减后保存到寄存器 x0 中
```

- **and**： 将某一寄存器的值和另一寄存器的值 按位与 并将结果保存到另一寄存器中，如：

```
and x0, x0, #0x1        将寄存器 x0 的值和常量 1 按位与后保存到寄存器 x0 中
```

- **orr**： 将某一寄存器的值和另一寄存器的值 按位或 并将结果保存到另一寄存器中，如：

```
orr x0, x0, #0x1          将寄存器 x0 的值和常量 1 按位或后保存到寄存器 x0 中
```

- **str** : 将寄存器中的值写入到内存中，如：

```
str x0, [x0, x8]     ; 将寄存器 x0 中的值保存到栈内存 [x0 + x8] 处
```

- **ldr**： 将内存中的值读取到寄存器中，如:

```
ldr x0, [x1, x2]  将寄存器 x1 和寄存器 x2 的值相加作为地址，取该内存地址的值放入寄存器 x0 中
```

- **cbz**： 和 0 比较，如果结果为零就转移（只能跳到后面的指令）
- **cbnz**： 和非 0 比较，如果结果非零就转移（只能跳到后面的指令）
- **cmp**： 比较指令
- **b**： （branch）跳转到某地址（无返回）
- **bl**： 跳转到某地址（有返回）
- **ret**： 子程序（函数调用）返回指令，返回地址已默认保存在寄存器 lr (x30) 中