

September 27, 2021

- C code \rightarrow ARM assembly (1)
 - comp. arch. ADDER (3)
 - rock/scissor/paper code \rightarrow codeblocks (2)
 - create library
-

Take fib.c \rightarrow ARM

1	2	3	4	5	6	7	8
1	1	2	3	5	8	13	21

fib.c

```
int fib(int nth)
{
    int m=1, n=1;
    if (nth==1)
        return 1;
    if (nth==2)
        return 1;

    for (int i=3;
        {
            m = n + m;
            n = m - n;
        }
    return m;
}
```

$\frac{n}{1} \quad \frac{m}{1} \quad (n+m) \quad (n+m)$

\downarrow

$m = n + m$

$n = (m - n)$

```

int fib (int nth)
{
    int m = 1, n = 1;
    if (nth == 1)
        goto quit;
    if (nth == 2)
        goto quit;
    m = 1;
    n = 1;
    while (1)
    {
        m = n;
        n = m + n;
        nth--;
        if (nth == 1)
            goto quit;
    }
    return m;
}

```

```
int i;  
i = 3;  
loop:  
    if (i > nth)  
        goto quit;  
    m = k + m;  
    n = m - n;  
    i = i + 1;  
    goto loop;  
quit;
```

quit: return m; }

ARM

```

int fib (int nth)
{ int n=1, m=1; ①
  if (nth == 1)
  { m=1 ②
    goto quit; }
  if (nth == 2)
  { m=1 ③
    goto quit; }
  int i; ④
  i = 3;
loop: ⑤
  if (i > nth) ⑥
    goto quit;
  m = m + n; ⑦
  n = m - n; ⑧
  i = i + 1; ⑨
  goto loop; ⑩
quit: return m; } ⑪

```

header code. fib.s

fib:

```

push {fp, lr}
add fp, sp, #4
@ r0 contains nth; r0 = nth
@ use r10 = n and r9 = m
mov r10, #1 @ n=1
mov r9, #1 @ m=1
①
cmp r0, #1
② beq quit
cmp r0, #2
③ beq quit
@ i = r8
④ mov r8, #3 @ i = 3
⑤ loop:
  cmp r8, r0
  ⑥ bgt quit

```

@ m = m + n ⑦

add r9, r9, r10 @ m = m + n

@ n = m - n ⑧

sub r10, r9, r10 @ n = m - n

@ i = i + 1 ⑨

add r8, r8, #1 @ i = i + 1

⑩ b loop @ goto loop

⑪ quit:

⑫ @ return m

@ m → r0

mov r0, r9 @ return m

⑬ sub sp, fp, #4
pop {fp, pc}