

- ✓ gdb with ARM sorted out
- ✓ week08 examples
  - ✓ fib-rec.s, sum-array-rec.s, substr-s.s
- ✓ project03 requirements
- ✓ Project03 questions
- ✓ ARM32 calling conventions

### ARM Instruction Set Format

ARM Reference  
"Machine code"  
Decoding

gdb

compile and assemble with

-g      ↴ do not args

gdb programe

gdb commands

"l" "list" list code

l function name

"b" breakpoints

b function name

b label

b line number

"r" "run"

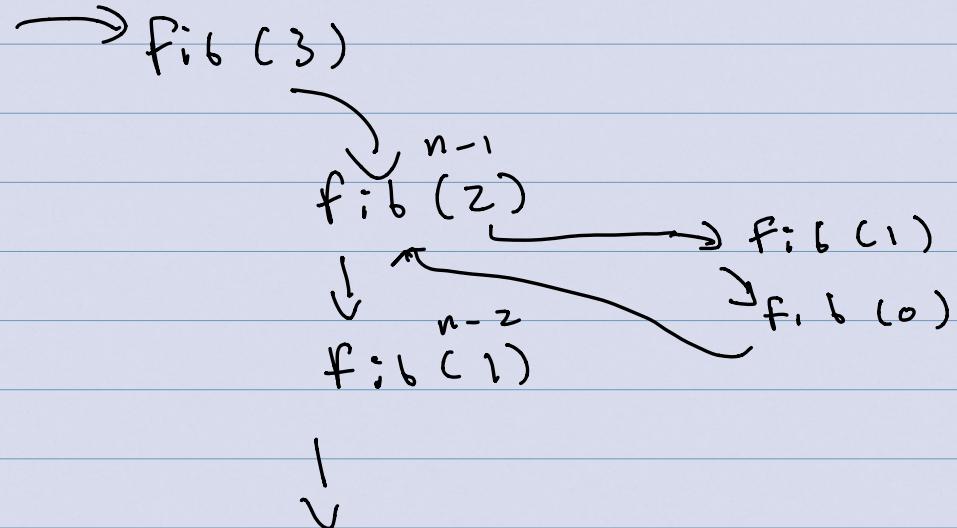
r arguments

refresh or CTRL-L to clean up  
display

"n" next → fine for C statements  
not good for assembly

"s" step "step;" → step → s

"c" continue → resumes execution  
to next breakpoint  
or end of program

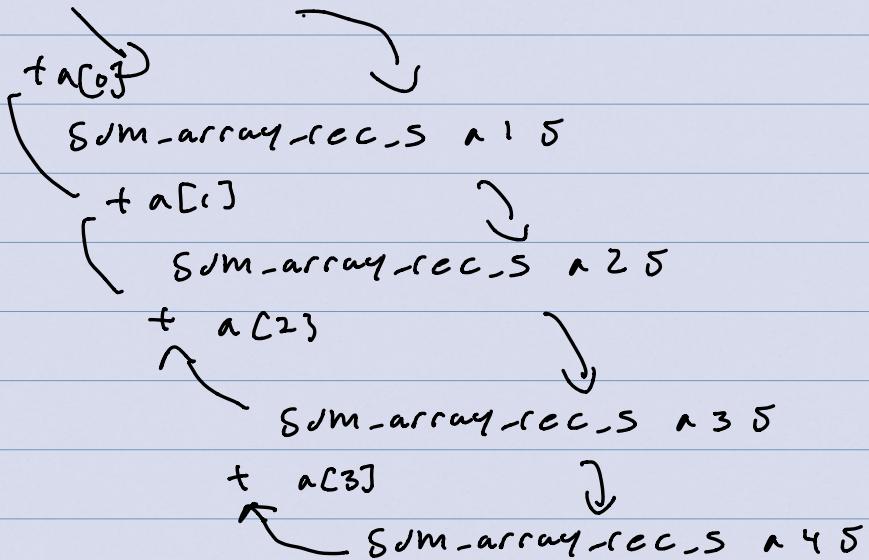


$f_{i6}(z)$

$f: \mathbb{C} \rightarrow \mathbb{C}$

$\text{fib}(0)$

sdm-array-rec-s n 05



## Project03 Requirements

Makefile ← week08/lecture/sec02

\$ make

⇒ project03

\$ make clean

project03.c ←

main

parse args

Files:

Makefile

project03.c

reverse.c.c

reverse-S.S

search.c.c

search-S.S

sort.c.c

sort-S.S

./project03 reverse "Hello world"

./project03 search 4 1 2 4 3 5



128 element

./project03 sort 1 10 2 11 3



lab06.c

fork/exec

# ARM Machine Code

$r_0 = r_0 + r_1 + 1$

add\_s:

$\text{add } r_0, r_0, r_1 \rightarrow$  [ ]

$\text{add } r_0, r_0, \#1 \rightarrow$  [ ]

$\text{bx } lr \rightarrow$  [ ]



32 bit  
instruction  
words

beq

add r0, r0, r1

0xE0800001

E	0	8	0	6	0	0	1
1110	0000	1000	0000	0000	0000	0000	0001
cond	00 I	opcode 6	rn	ro	rm		

E AL      0100 add      r1

~~add r0, r0, r1~~

add r0, r0, r1