Outline

- FP Q&A
- Conventions, contracts & protocols
- Starting a program,
- Turning on a machine

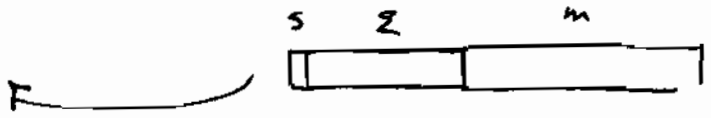
I FP Q&A - why

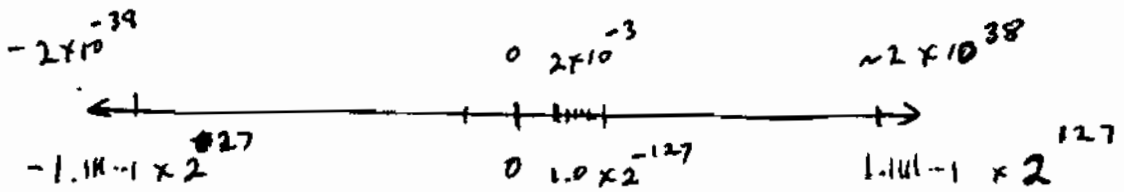
Q. what does a computer do ?
- manipulates symbols

* Big idea *

Q: n bits $\Rightarrow$ 2^n symbols
Q: what do they mean ?
- depends what you do with them

$-1^S \times 1.M \times 2^{E-127}$





Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.___

Special Symbols

denorm

E = 0 $\quad 0, \pm 1.0.M \times 2^{-127}$

E = 255 $\quad +\infty, -\infty$

NAN

Why

$A \neq B \Rightarrow A - B \neq 0$

trap $\quad X / A-B$

If X is denorm, what is $1/x$? $\quad \pm\infty$

Q: Why does it matter?

— knowing where the illusion breaks down ⇒ good engineering

ex: roots of a quadratic

II Conventions

PROG LANG → (convention) → ASSM LANG

$1 Asm temp

$2-$3 $V0, $1

4-7 $a0-$a3

8-15 $t0-$t7

16-23

- $0
- $AT
- $V0, V1
- $a0-a3
- 9 $t0-t7 — I promise not to put anything important
- 16 $s0-s7 — you promise to preserve
- $t8-t9
- $k0-$k1 — kernel
- $GP → 0x10008000
- $SP → 0x7FFFEFFC
- $FP
- $RA

PC → 0x00400000

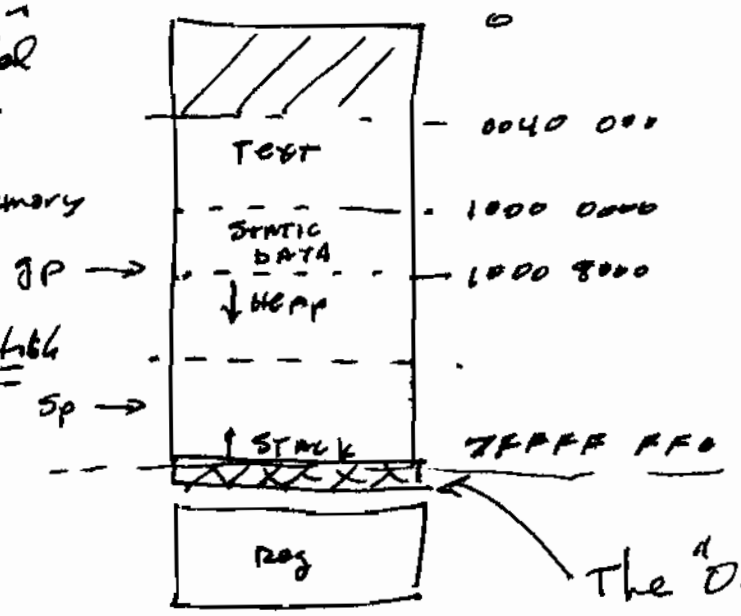
Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.__

Starting a program

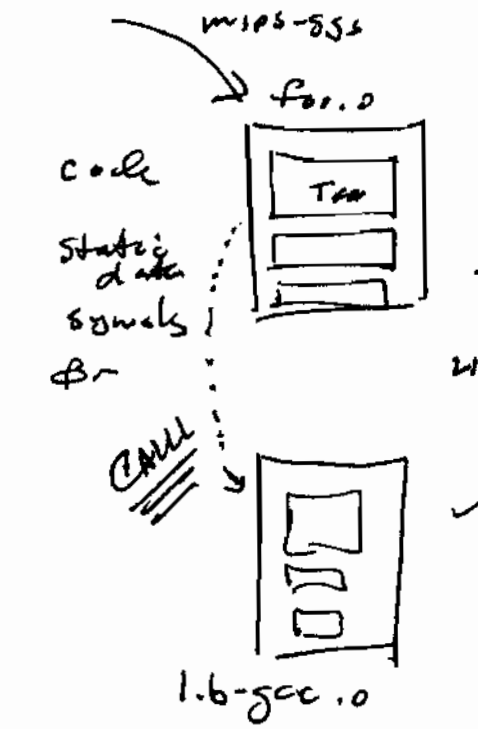
- A special program is "always running" called the operating system
- Reads a file into memory
  Jumps to start instruction
  - Header of executable
- Language RT calls MAIN

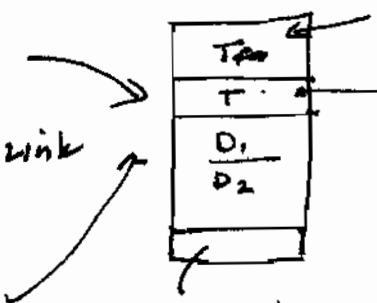
Today: physical address space
Later: virtual address space
" the memory abstraction



The "Other half"

foo.c





CRT0 jal main

main: ← What the assembler writes

- Registers
- Address space
  - sp, gp
- Run Time data structures
  - malloc
  - arg c
  - arg v
  - stdin, stdout, stderr

CRT0
"--main"
- Tiny bit of assembly
- System calls

To Page No.___

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

rom Page No.__

Q: What function do you call?
- pgm know the name.
- machine needs an address
- ~~them~~ not known until all the objects are put together
- linking resolves the ~~addr~~

* like what you did with forward references in assembler

Language run time

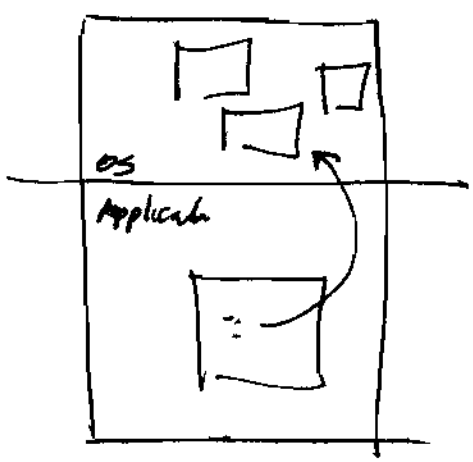
- sets up machine state to meet ~~Lang~~ language conventions
- sets up data structures and application program state

⇒ System calls.

HW support

mode

2PC



OS: • provides a convenient abstraction of physical resources
- disks ⇒ files
• Protected access to shared resources

How does appln "call" a system function
- don't have address of routine
"separate" address space
- system protects itself
- not an user ~~stack~~ - return address

To Page No.____

Witnessed & Understood by me	Date	Invented by	Date
		Recorded by	

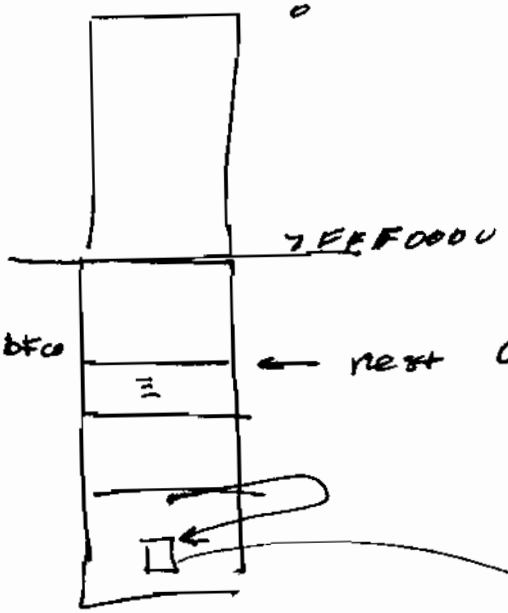
From Page No.

3 ~~kinds~~ kinds of entries

Traps (synchronous to inst) → syscall, exception

Interrupts

MIPS



0x bfc0 0000

4	0200	TLB
4	0300	Cache error
11	0380	other

General form PC = 0x bfc0

code to determine the cause of the trap or interrupt & service it

Syscall

Place arguments in registers
execute syscall inst → OS int vector

- system code has access to HW resource that are not accessible to application

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date	
		Recorded by		

From Page No.

What happens when you turn it on?

PC ← 0xbFC0 0000

registers undefined

⇒ OS boot strap
- verify hw is operational
- bring up all the subsystems
- start process schedr

How did OS get there
- prom e.g. BIOS
- boot loader – reads blocks in from disk

Witnessed & Understood by me,	Date	Invented by	Date	
		Recorded by		