Calling conventions and GDB

(aka GDB is not that scary and actually useful)

GDB

- Run GDB locally
- Run GDB on Athena
- Questions about GDB

Refresher on assembly

- C -> assembly (.S or .asm files) -> binary (.o files)
- https://6190.mit.edu/_static/fall22/resources/references/riscv_isa_reference.pdf
- Demo 1

RISC-V vs x86

- RISC-V specifications
- x86 specifications
- Other reduced instruction sets: https://en.wikipedia.org/wiki/List_of_products_using_ARM_processors

Calling conventions

Register	ABI Name	Description	Saver
x0	zero	Hard-wired zero	—
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5–7	t0–2	Temporaries	Caller
x8	s0/fp	Saved register/frame pointer	Callee
x9	s1	Saved register	Callee
x10–11	a0–1	Function arguments/return values	Caller
x12–17	a2–7	Function arguments	Caller
x18–27	s2–11	Saved registers	Callee
x28–31	t3–6	Temporaries	Caller
f0–7	ft0–7	FP temporaries	Caller
f8–9	fs0–1	FP saved registers	Callee
f10–11	fa0–1	FP arguments/return values	Caller
f12–17	fa2–7	FP arguments	Caller
f18–27	fs2–11	FP saved registers	Callee
f28–31	ft8–11	FP temporaries	Caller

Table 18.2: RISC-V calling convention register usage.

What happens if we don’t follow calling conventions?

Show demo 2

Virtual address space

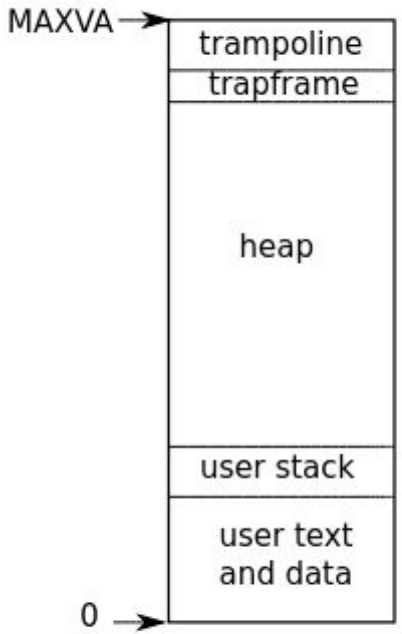
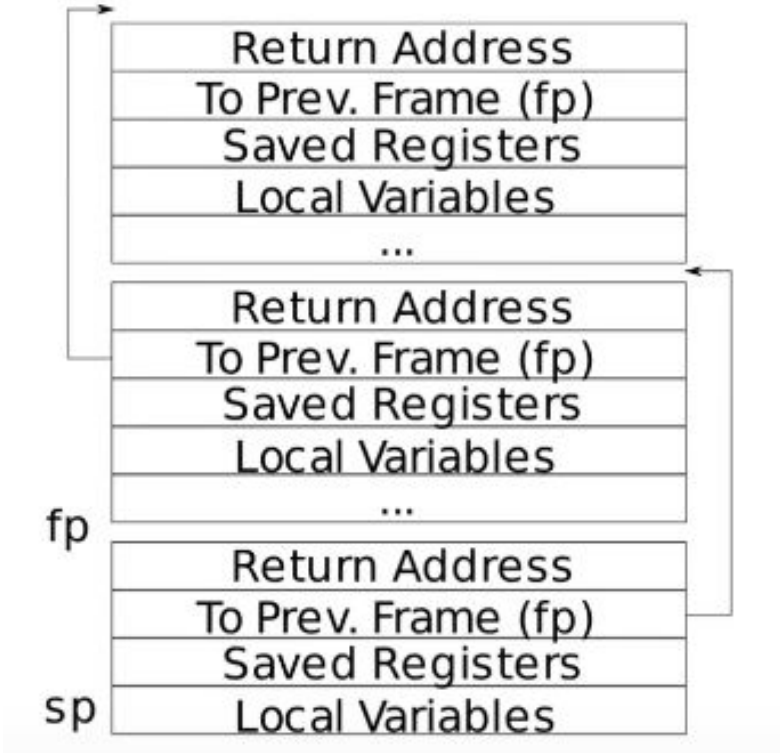


Figure 2.3: Layout of a process's virtual address space

The stack


Return Address
To Prev. Frame (fp)
Saved Registers
Local Variables
...
Return Address
To Prev. Frame (fp)
Saved Registers
Local Variables
...
fp
Return Address
To Prev. Frame (fp)
Saved Registers
sp
Local Variables

More demos!

Show demo 3, 4 and 5