CS61 Lecture 10 — October 3

Ex Program 1:

```
        pushq   %r12
        pushq   %rbp
        pushq   %rbx
        testl   %edi, %edi
        je      .L4
        movl    %edi, %r12d
        movl    $0, %ebx
        movl    $0, %ebp

.L4
        movl    %edi, %ebp
        jmp     .L1

.L3
        movl    %ebx, %edi
        call    _Z1gj@PLT
        addl    %eax, %ebp
        addl    $1, %ebx
        cmpl    %ebx, %r12d
        jne     .L3

.L1
        movl    %ebp, %eax
        popq    %rbx
        popq    %rbp
        popq    %r12
        ret
```

register allocation

- turned off w/ allocation

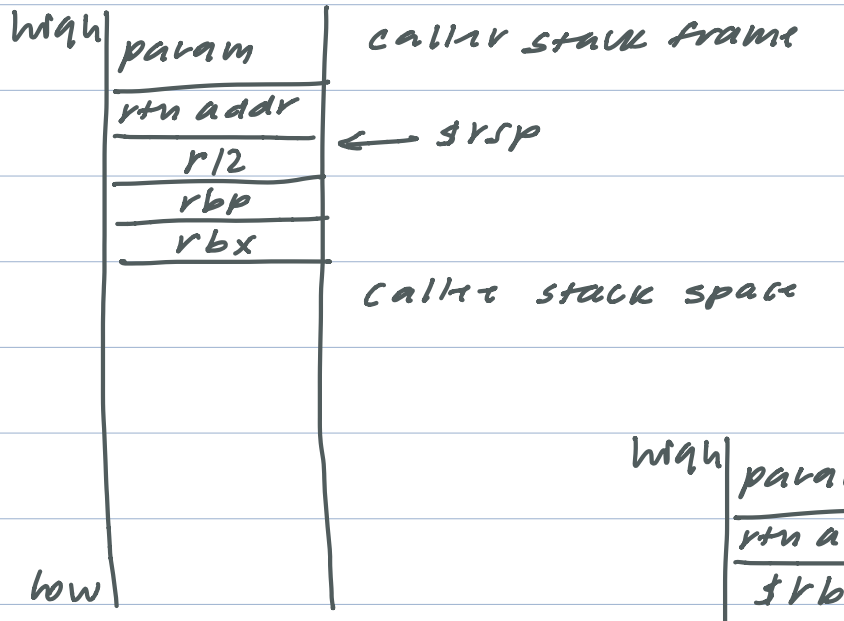
Callee-saved registers
Caller-owned registers
} equivalent, where value is maintained

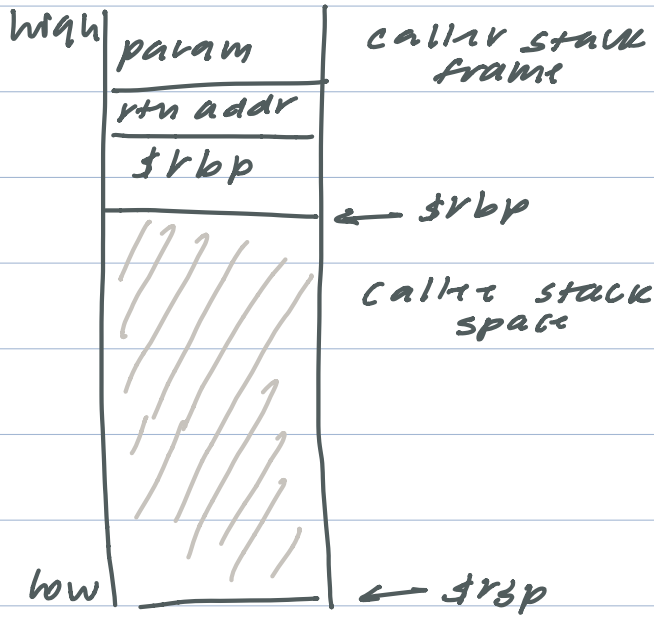
↳ keep saved values in the stack

Caller-saved register
Callee-owned register
} equivalent, where called function can do whatever it wants w/ register

↳ ex: rdi

Stack





$rbp ← $rsp

EX Program 2 (bomb)

sub $0x8, %rsp : alignment

objdump -t big-boom | less

procedure inlining

tail call elimination

Comparison

mov (%rdi), %eax

lea 0x4(%rsi), %eax ↔ same for 4-bit object!

argument register: rdi

arithmetic operations: mess w/ flags
pointer arithmatic does not

- familiarity of codebase, finish phase 1