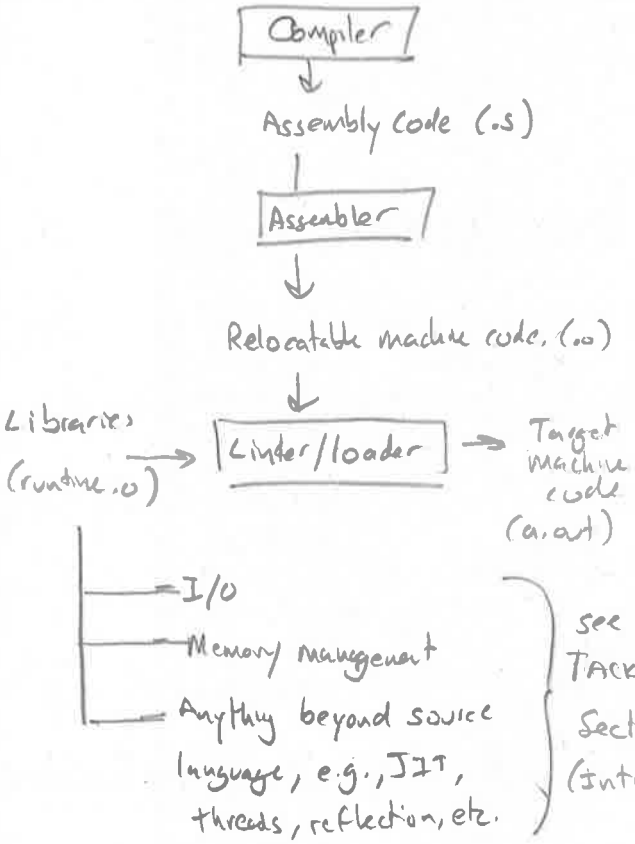
Runtime environments pg. 1

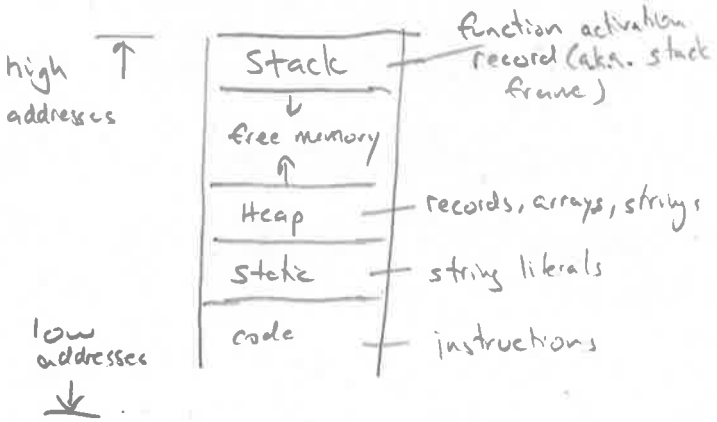
Lecture topics

I. Runtime Systems
II Stack
III Heap

I. Runtime Systems



Storage organization



II Stack

```
main { f(); g(); }
f{ if .... f(); else g(); }
g { ... }
```

Example TACK

$x = f(E_0, \ldots, E_{n-1})$

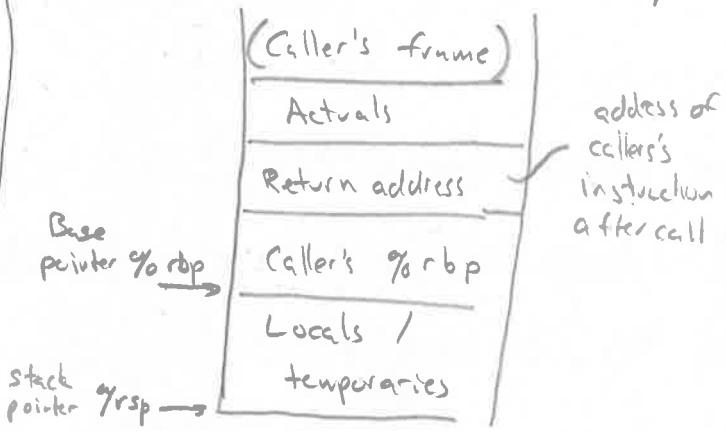
IR

E_0.code
...
E_{n-1}.code
param[0:n] = E_0.addr
...
param[n-1:n] = E_{n-1}.addr
x = call f:n

→ E;

E.code
return E.addr

Stack frame layout (see x64-intro.pdf)



Call sequence

- put actuals in register/stack
- call (pushes return address)

} caller

- push %rbp
- move %rbp, %rsp
- sub %rsp, frame size

} callee

Return sequence

- move %rax, return value
- move %rsp, %rbp
- pop %rbp
- ret (pops return address)

} callee

- remove actuals from stack
- use %rax

} caller

Other stack topics:

- saving registers across calls
- variable size data on stack
- accessing non-local stack variables

III Heap

```
x = (a=1);
y = (a=2);
x = null
→ y;
```

lifetimes of heap objects are not nested, may be indefinite

	Java	C	C++	Tack
Allocation	new	malloc	new	records/array literals
Deallocation	garbage collection	free	delete	unspecified