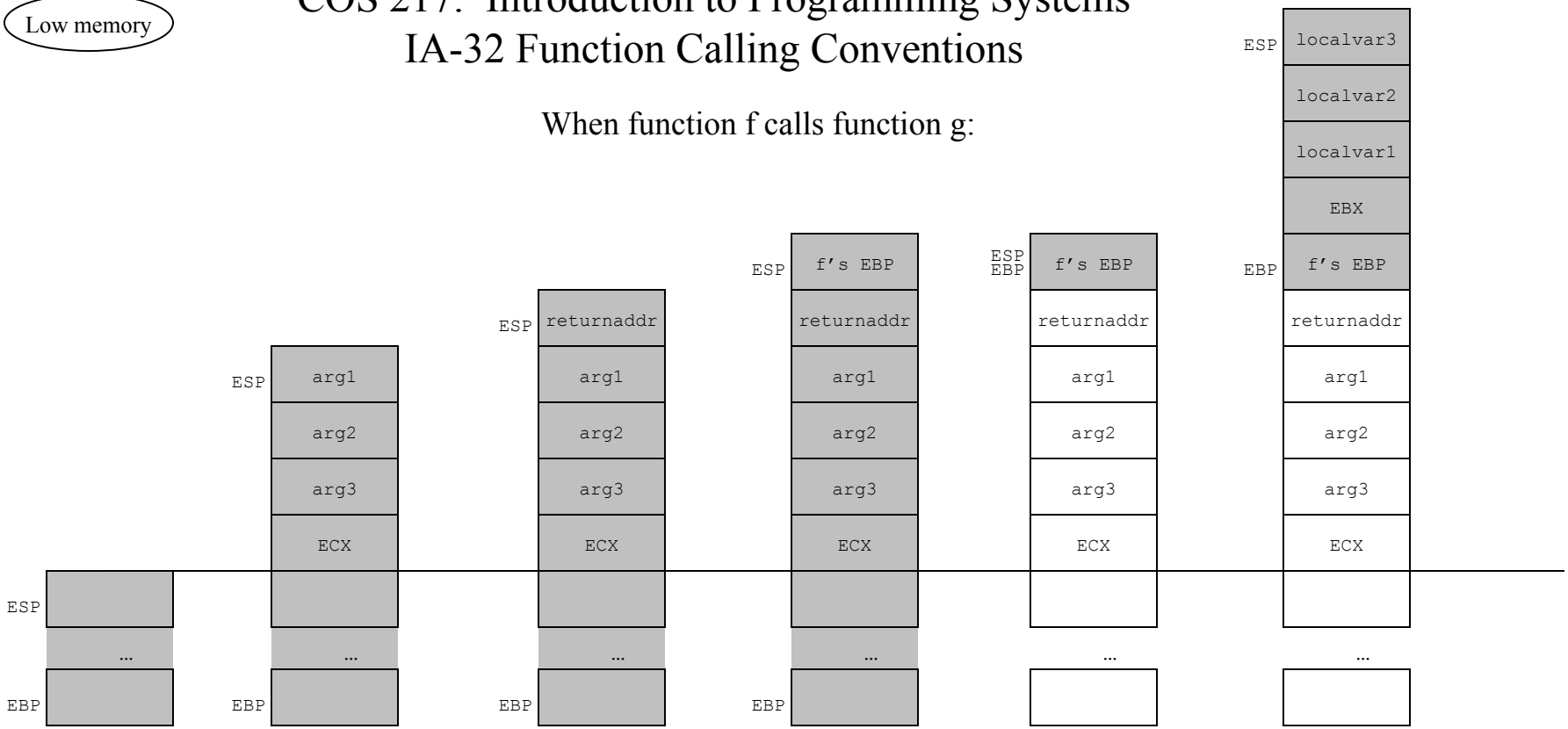
Princeton University
COS 217: Introduction to Programming Systems
IA-32 Function Calling Conventions

Low memory

When function f calls function g:



(1) If necessary, f pushes EAX, ECX, EDX, and arguments

```
pushl %ecx
pushl arg3
pushl arg2
pushl arg1
```

(2) f executes call instruction:

```
call g
```

(3) g pushes f's EBP:

```
pushl %ebp
```

(4) g copies ESP to EBP:

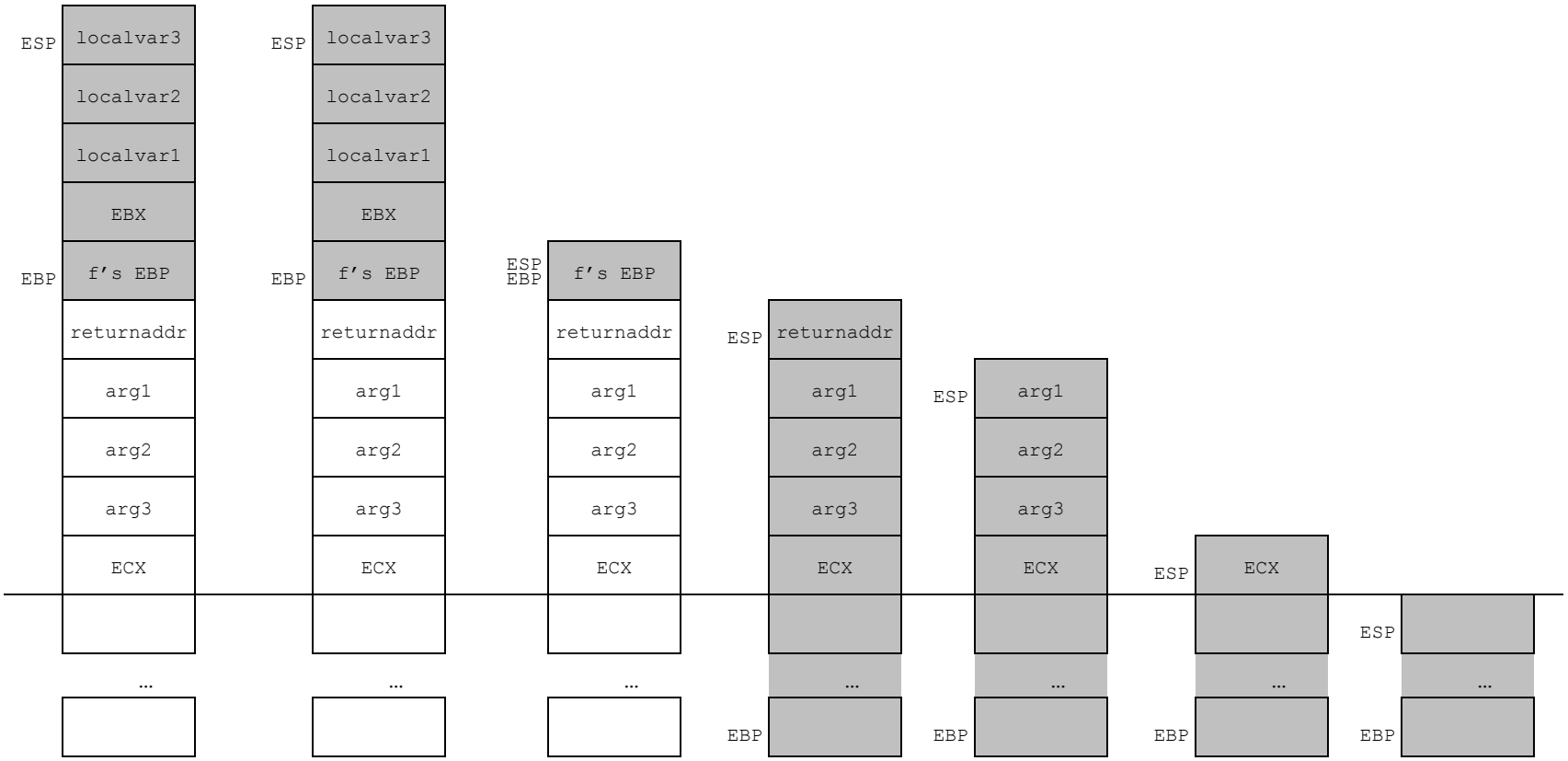
```
movl %esp,%ebp
```

Prolog

(5) If necessary, g pushes EBX, ESI, EDI and local variables:

```
pushl %ebx
pushl localvar1
pushl localvar2
pushl localvar3
```

High memory



(6) g uses its arguments (now called parameters) and local variables to compute a return value, and moves that value to EAX:

```
movl 8(%ebp), ???
movl 12(%ebp), ???
…
movl -8(%ebp), ???
movl -12(%ebp), ???
…
movl ???, %eax
```

(7) If necessary, g restores EBX, ESI, EDI

```
movl -4(%ebp),%ebx
```

(8) g copies EBP to ESP:

```
movl %ebp,%esp
```

(9) g pops from stack into EBP:

```
popl %ebp
```

Epilog

(10) g executes ret instruction:

```
ret
```

(11) f pops arguments from stack, and uses return value in EAX:

```
addl $12, %esp
movl %eax, ???
```

(12) If necessary, f restores EAX, ECX, EDX

```
popl %ecx
```