

# 운영체제의 기초: Stack and Dynamic Memory Allocation of Local Variables

2023년 3월 21일

홍성수

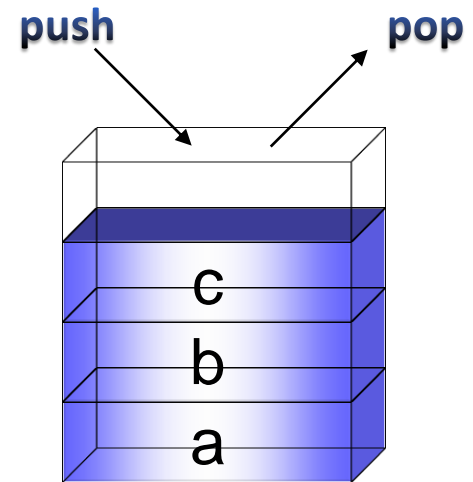
[sshong@redwood.snu.ac.kr](mailto:sshong@redwood.snu.ac.kr)

SNU RTOSLab 지도교수  
서울대학교 전기정보공학부 교수

## What is Stack?

### ❖ 스택 (stack)

- “나중에 들어간 원소가 먼저 나오는 원칙”에 따라 데이터의 삽입과 삭제가 이루어지는 자료 구조
  - Last-in-first-out (LIFO)
  - (예) 함수 호출, 수식 계산, 변수의 할당과 반환
  - push(): 스택에 한 원소를 삽입
  - pop(): 스택에서 한 원소를 삭제
    - (예) push(a)→push(b) → push(c) → pop()



## 스택을 이용한 변수 할당 (1)

```
main()
{
    int a = 2, b = 13;
    int res;
    res = add(a, b);
}
```

```
int add(int x, int y)
{
    int r;
    r = x + y;
    return r;
}
```

main:

add:

```
pushl %ebp
movl %esp,%ebp
subl $4,%esp
movl 8(%ebp),%eax
movl 12(%ebp),%edx
leal (%edx,%eax),%ecx
movl %ecx,-4(%ebp)
movl -4(%ebp),%edx
movl %edx,%eax
jmp .L2
.align
```

.L1:

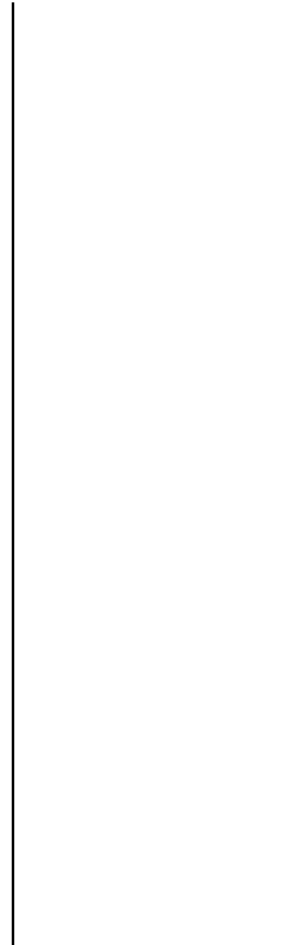
.L2:

```
movl %ebp,%esp
popl %ebp
ret
```

## 스택을 이용한 변수 할당 (2)

main:

```
pushl %ebp
movl %esp,%ebp
subl $12,%esp
movl $2,-4(%ebp)
movl $13,-8(%ebp)
movl -8(%ebp),%eax
pushl %eax
movl -4(%ebp),%eax
pushl %eax
call add
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```

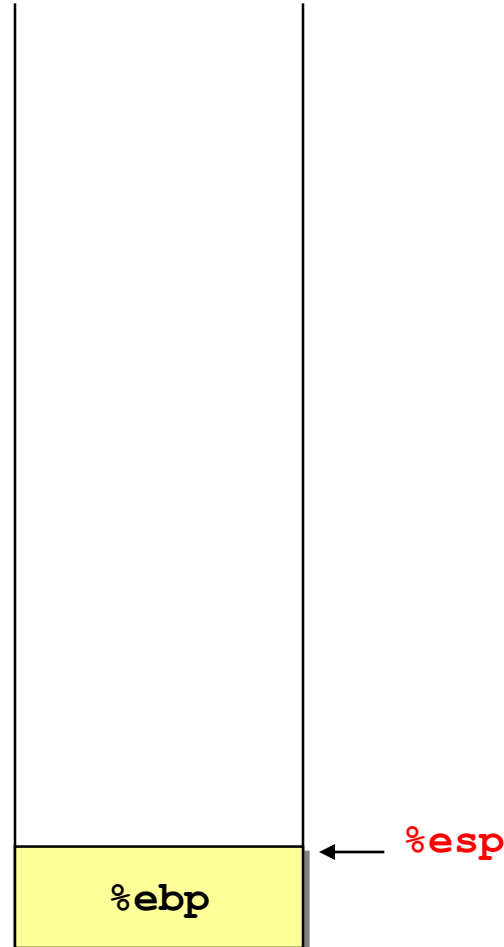


← %esp Seoul National University

## 스택을 이용한 변수 할당 (2)

main:

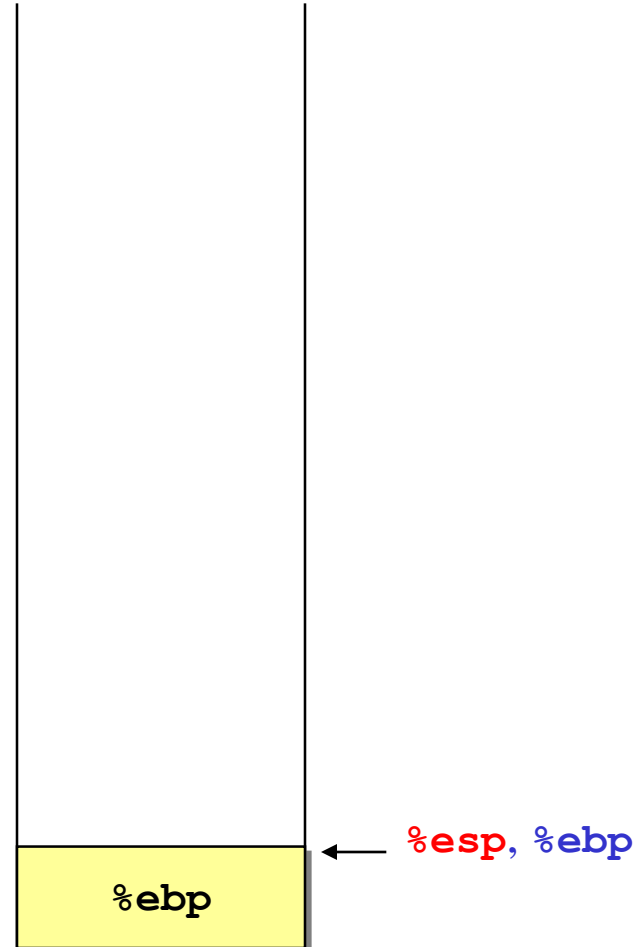
```
→ pushl %ebp
   movl %esp,%ebp
   subl $12,%esp
   movl $2,-4(%ebp)
   movl $13,-8(%ebp)
   movl -8(%ebp),%eax
   pushl %eax
   movl -4(%ebp),%eax
   pushl %eax
   call add
   addl $8,%esp
   movl %eax,%eax
   movl %eax,-12(%ebp)
```



## 스택을 이용한 변수 할당 (2)

main:

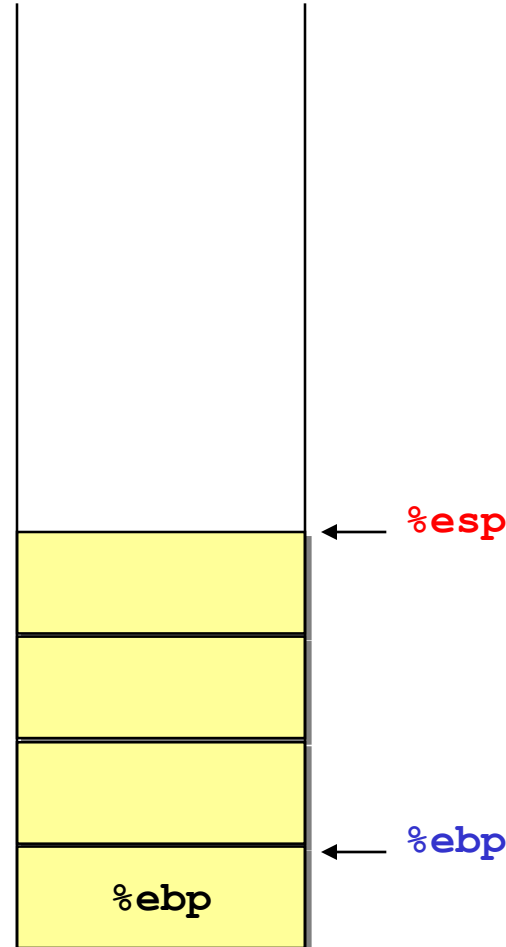
```
    pushl %ebp  
    → movl %esp,%ebp  
      subl $12,%esp  
      movl $2,-4(%ebp)  
      movl $13,-8(%ebp)  
      movl -8(%ebp),%eax  
      pushl %eax  
      movl -4(%ebp),%eax  
      pushl %eax  
      call add  
      addl $8,%esp  
      movl %eax,%eax  
      movl %eax,-12(%ebp)
```



## 스택을 이용한 변수 할당 (2)

main:

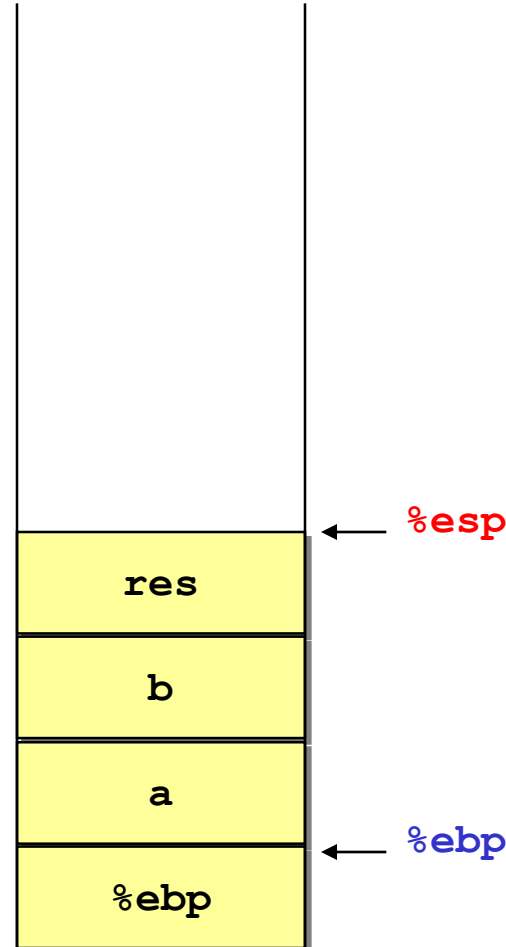
```
pushl %ebp
movl %esp,%ebp
→ subl $12,%esp
movl $2,-4(%ebp)
movl $13,-8(%ebp)
movl -8(%ebp),%eax
pushl %eax
movl -4(%ebp),%eax
pushl %eax
call add
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```



## 스택을 이용한 변수 할당 (2)

main:

```
pushl %ebp
movl %esp,%ebp
→ subl $12,%esp
movl $2,-4(%ebp)
movl $13,-8(%ebp)
movl -8(%ebp),%eax
pushl %eax
movl -4(%ebp),%eax
pushl %eax
call add
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```

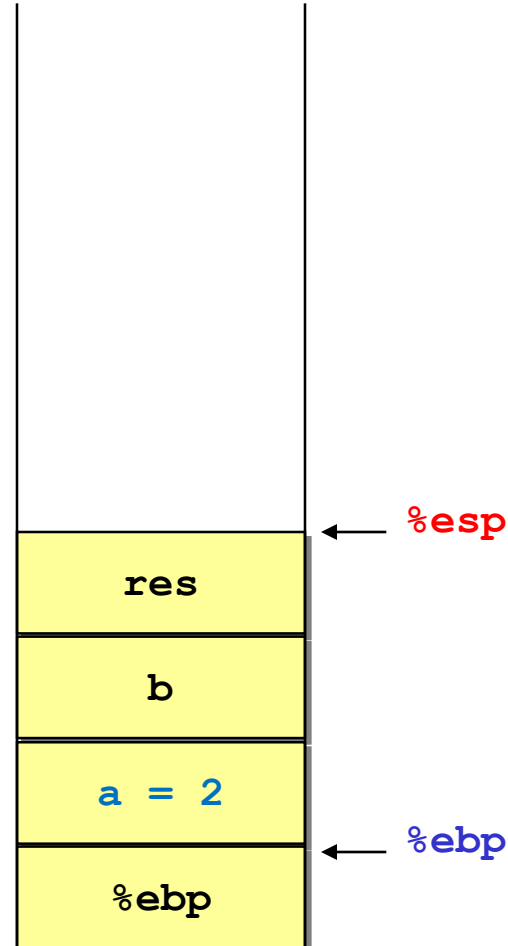




## 스택을 이용한 변수 할당 (2)

main:

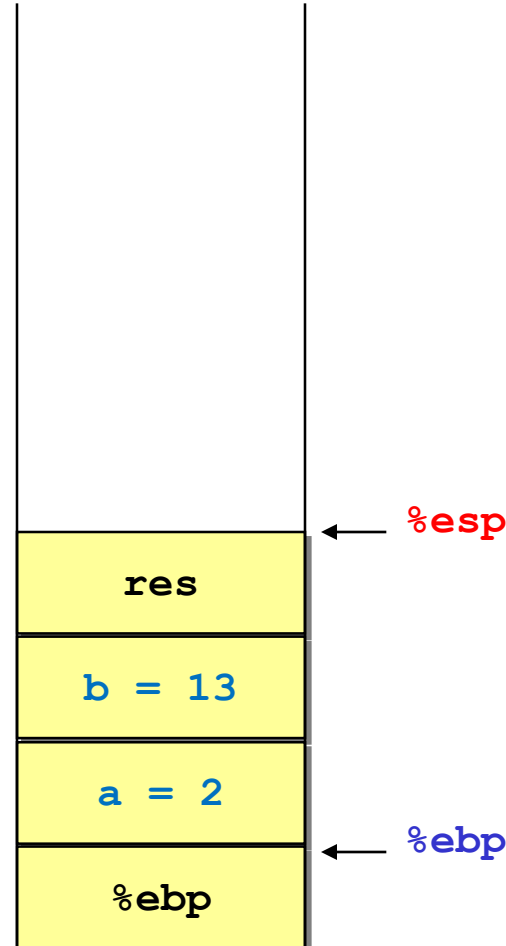
```
pushl %ebp
movl %esp,%ebp
subl $12,%esp
➔ movl $2,-4(%ebp)
movl $13,-8(%ebp)
movl -8(%ebp),%eax
pushl %eax
movl -4(%ebp),%eax
pushl %eax
call add
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```



## 스택을 이용한 변수 할당 (2)

main:

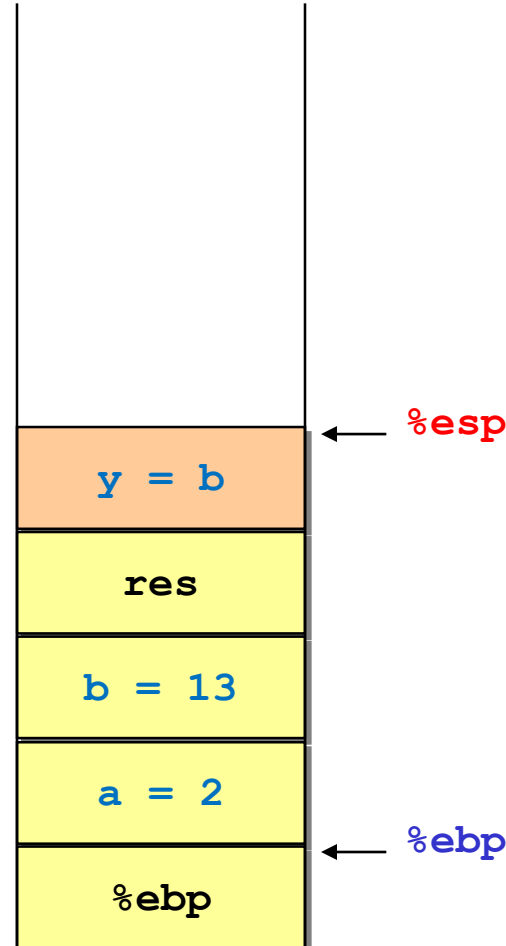
```
pushl %ebp
movl %esp,%ebp
subl $12,%esp
movl $2,-4(%ebp)
➔ movl $13,-8(%ebp)
movl -8(%ebp),%eax
pushl %eax
movl -4(%ebp),%eax
pushl %eax
call add
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```



## 스택을 이용한 변수 할당 (2)

main:

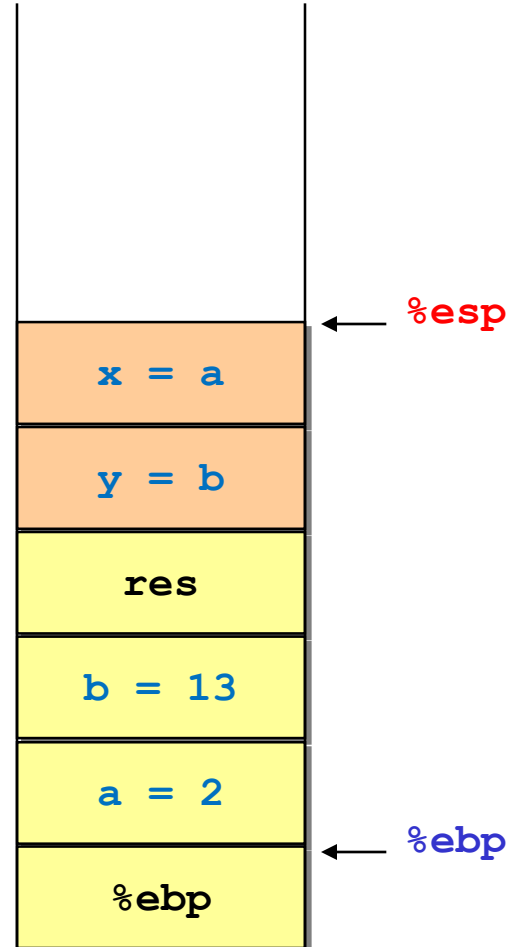
```
pushl %ebp
movl %esp,%ebp
subl $12,%esp
movl $2,-4(%ebp)
movl $13,-8(%ebp)
→ { movl -8(%ebp),%eax
    pushl %eax
    movl -4(%ebp),%eax
    pushl %eax
    call add
    addl $8,%esp
    movl %eax,%eax
    movl %eax,-12(%ebp)
```



## 스택을 이용한 변수 할당 (2)

main:

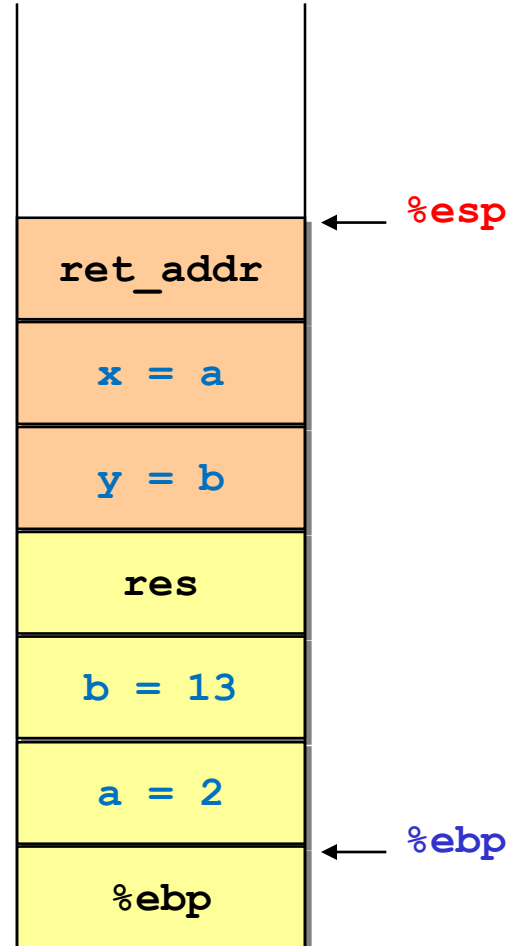
```
pushl %ebp
movl %esp,%ebp
subl $12,%esp
movl $2,-4(%ebp)
movl $13,-8(%ebp)
movl -8(%ebp),%eax
pushl %eax
→ { movl -4(%ebp),%eax
    pushl %eax
    call add
    addl $8,%esp
    movl %eax,%eax
    movl %eax,-12(%ebp)
```



## 스택을 이용한 변수 할당 (2)

main:

```
pushl %ebp
movl %esp,%ebp
subl $12,%esp
movl $2,-4(%ebp)
movl $13,-8(%ebp)
movl -8(%ebp),%eax
pushl %eax
movl -4(%ebp),%eax
pushl %eax
→ call add
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```



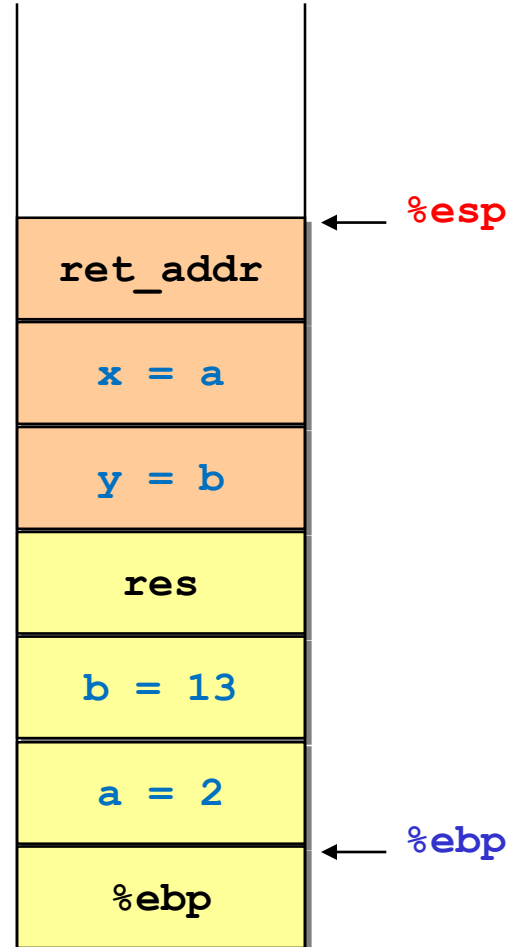
## 스택을 이용한 변수 할당 (3)

add:

```
pushl %ebp
movl %esp,%ebp
subl $4,%esp
movl 8(%ebp),%eax
movl 12(%ebp),%edx
leal (%edx,%eax),%ecx
movl %ecx,-4(%ebp)
movl -4(%ebp),%edx
movl %edx,%eax
jmp .L2
.align 4
```

.L2:

```
movl %ebp,%esp
popl %ebp
ret
```



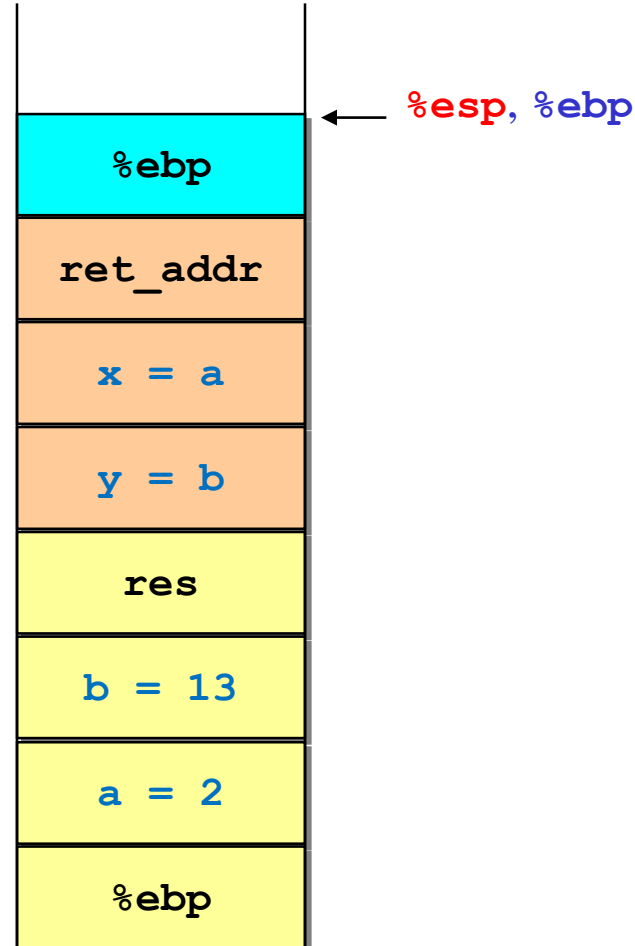
## 스택을 이용한 변수 할당 (3)

add:

```
→ { pushl %ebp  
    movl %esp,%ebp  
    subl $4,%esp  
    movl 8(%ebp),%eax  
    movl 12(%ebp),%edx  
    leal (%edx,%eax),%ecx  
    movl %ecx,-4(%ebp)  
    movl -4(%ebp),%edx  
    movl %edx,%eax  
    jmp .L2  
    .align 4
```

.L2:

```
    movl %ebp,%esp  
    popl %ebp  
    ret
```



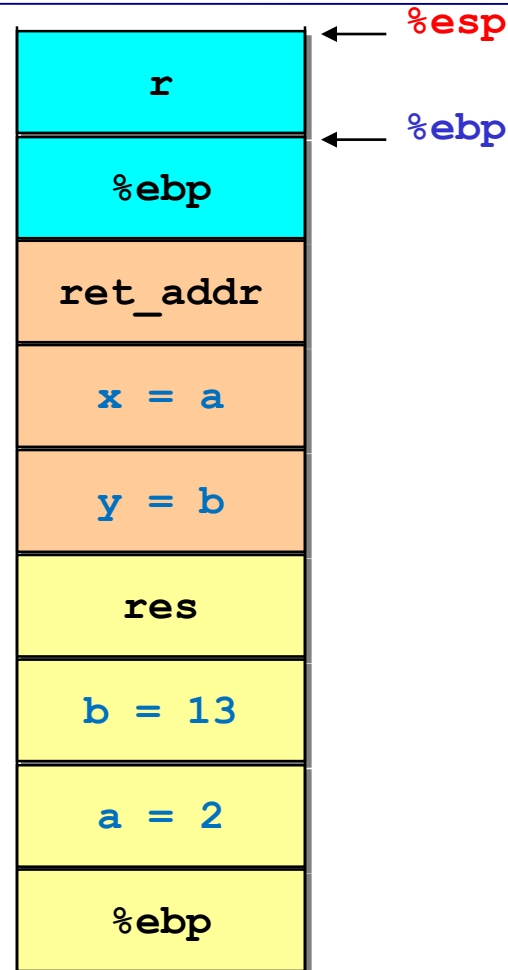
## 스택을 이용한 변수 할당 (3)

add:

```
pushl %ebp
movl %esp,%ebp
→ subl $4,%esp
movl 8(%ebp),%eax
movl 12(%ebp),%edx
leal (%edx,%eax),%ecx
movl %ecx,-4(%ebp)
movl -4(%ebp),%edx
movl %edx,%eax
jmp .L2
.align 4
```

.L2:

```
movl %ebp,%esp
popl %ebp
ret
```





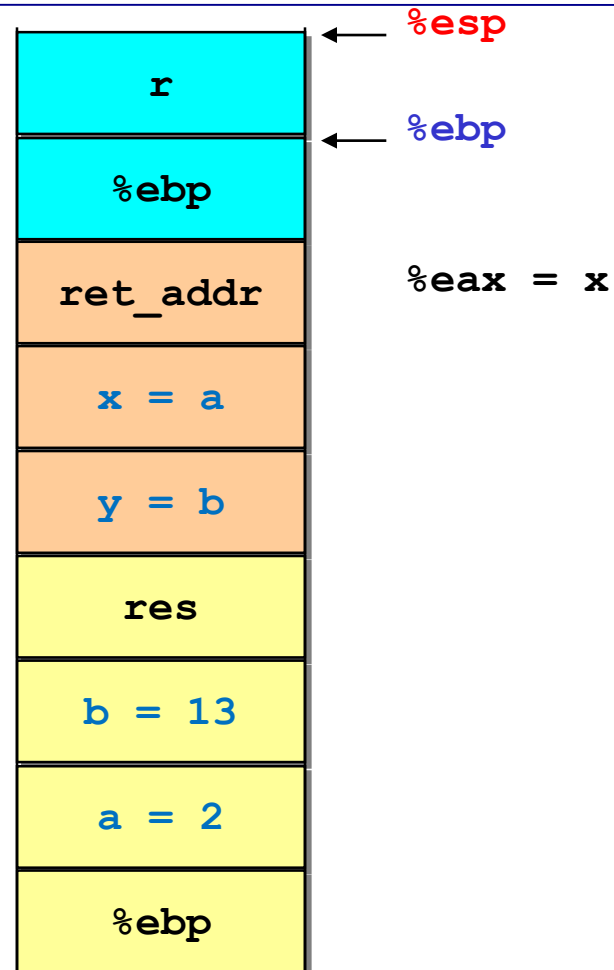
## 스택을 이용한 변수 할당 (3)

add:

```
pushl %ebp
movl %esp,%ebp
subl $4,%esp
➔ movl 8(%ebp),%eax
movl 12(%ebp),%edx
leal (%edx,%eax),%ecx
movl %ecx,-4(%ebp)
movl -4(%ebp),%edx
movl %edx,%eax
jmp .L2
.align 4
```

.L2:

```
movl %ebp,%esp
popl %ebp
ret
```



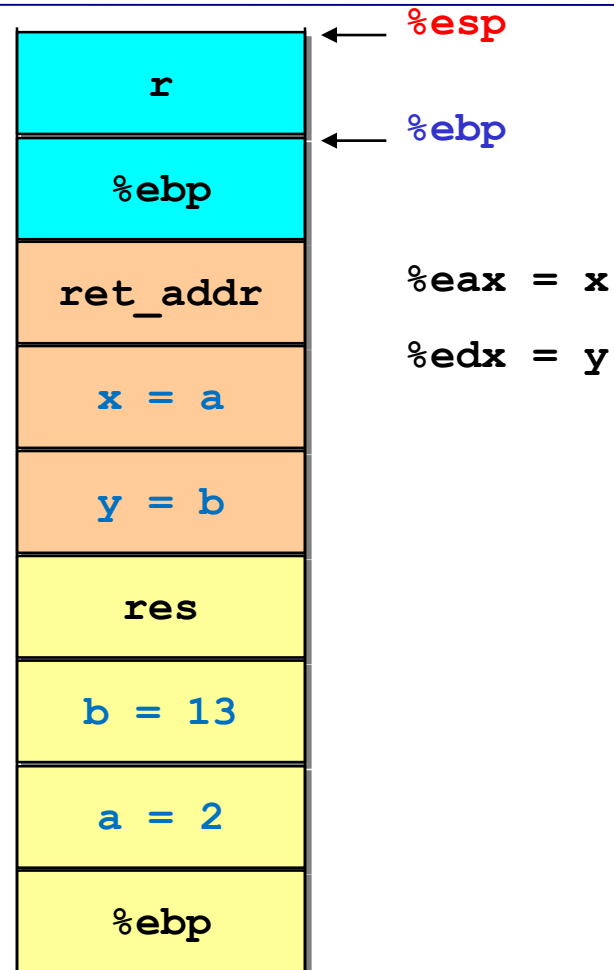
## 스택을 이용한 변수 할당 (3)

add:

```
pushl %ebp
movl %esp,%ebp
subl $4,%esp
movl 8(%ebp),%eax
➔ movl 12(%ebp),%edx
leal (%edx,%eax),%ecx
movl %ecx,-4(%ebp)
movl -4(%ebp),%edx
movl %edx,%eax
jmp .L2
.align 4
```

.L2:

```
movl %ebp,%esp
popl %ebp
ret
```



## 스택을 이용한 변수 할당 (3)

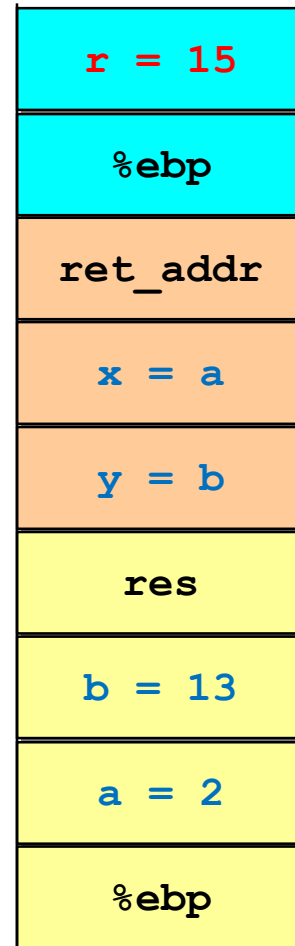
add:

```
pushl %ebp
movl %esp,%ebp
subl %4,%esp
movl 8(%ebp),%eax
movl 12(%ebp),%edx
leal (%edx,%eax),%ecx
movl %ecx,-4(%ebp)
movl -4(%ebp),%edx
movl %edx,%eax
jmp .L2
.align 4
```



.L2:

```
movl %ebp,%esp
popl %ebp
ret
```



`%esp`

`%ebp`

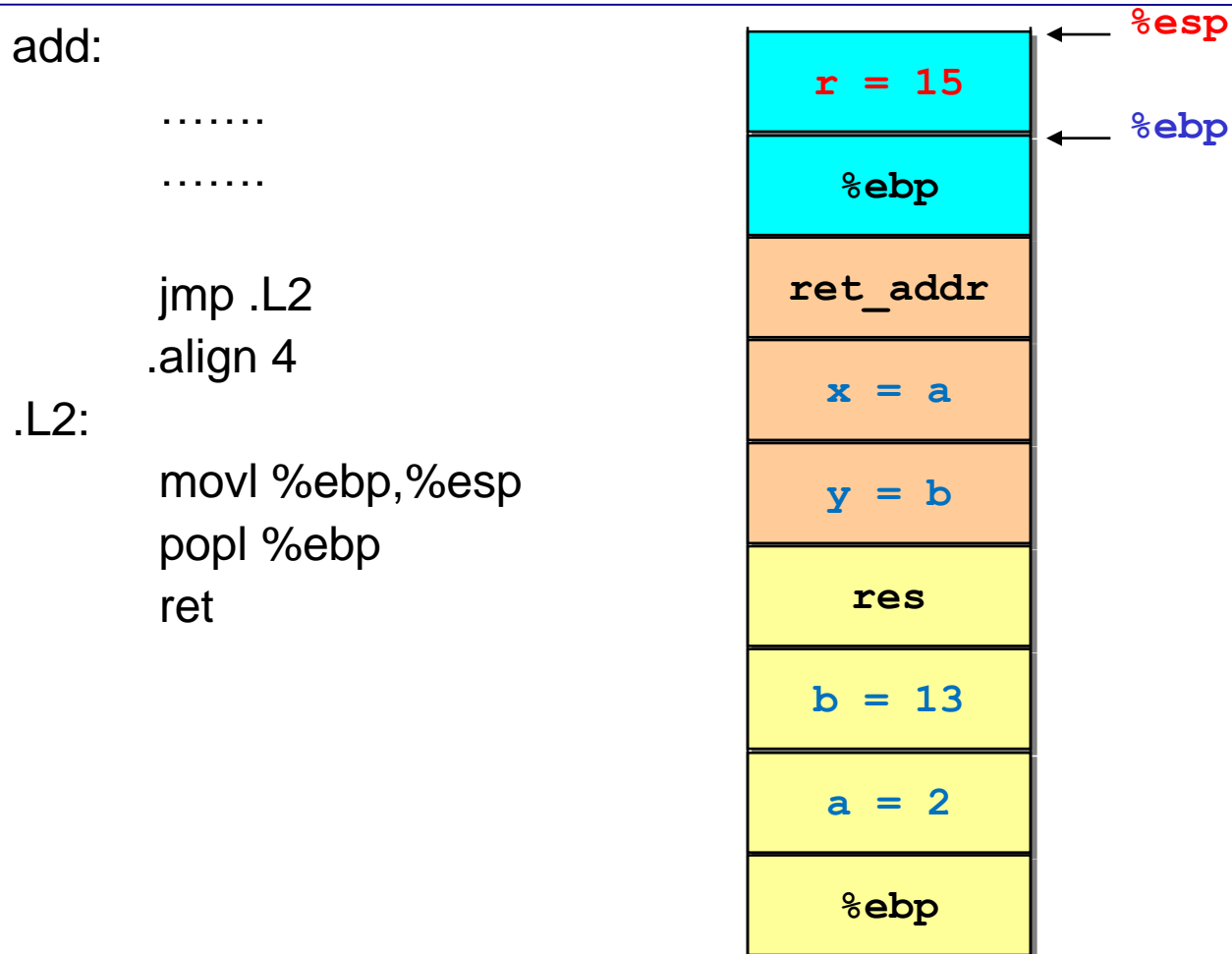
`%ecx = %edx + %eax`  
`D(%r1,%r2,S) == %r1 + %r2*S + D`

`r = %ecx`

`%edx = r`

`%eax = %edx`

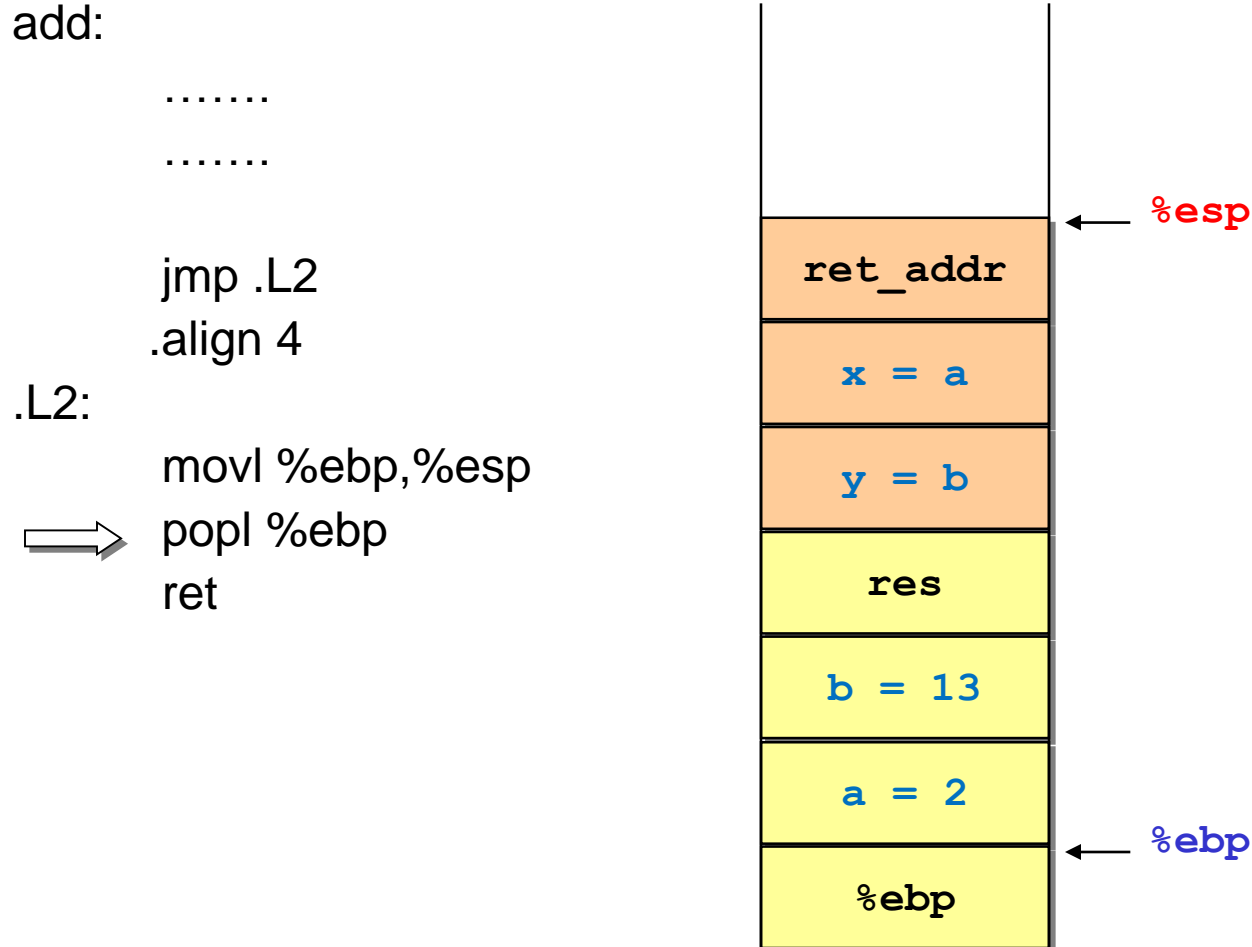
## 스택을 이용한 변수 할당 (4)



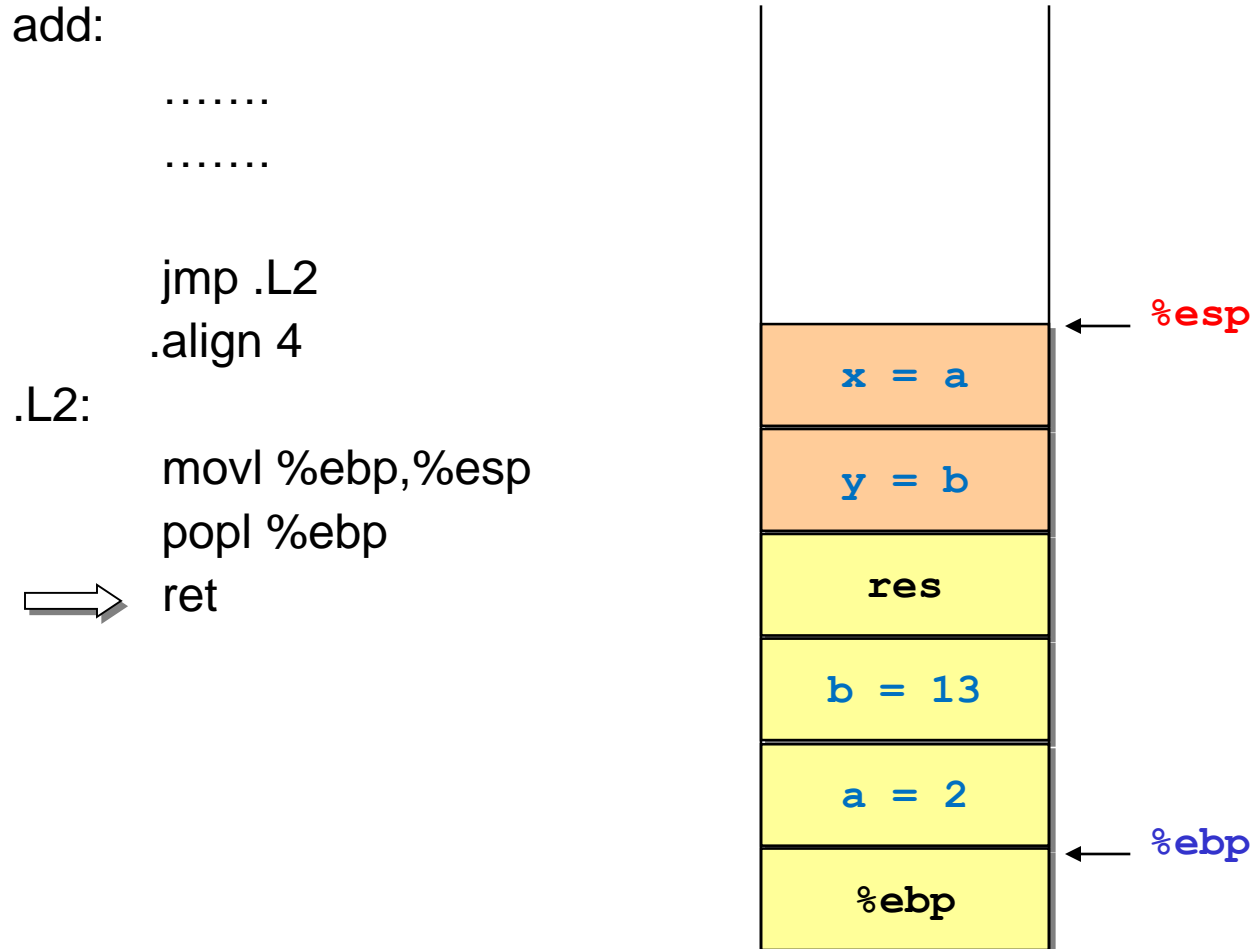
## 스택을 이용한 변수 할당 (4)



## 스택을 이용한 변수 할당 (4)



## 변수와 Stack (5)



## 스택을 이용한 변수 할당 (5)

main:

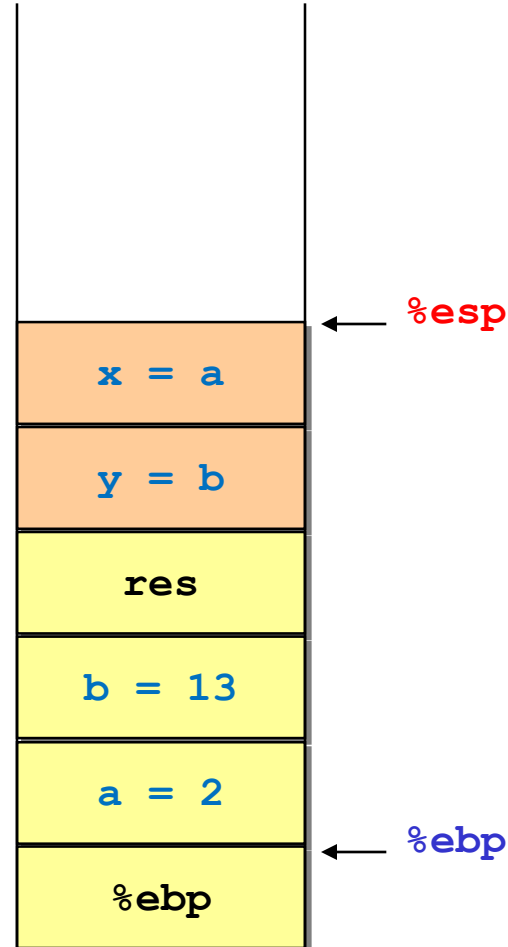
.....

.....

```
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```

.L1:

```
movl %ebp,%esp
popl %ebp
ret
```





## 스택을 이용한 변수 할당 (5)

main:

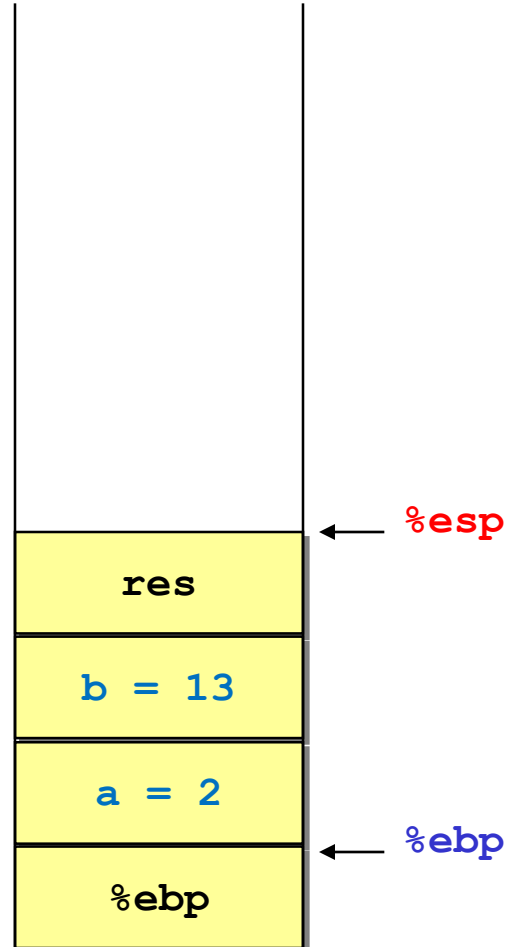
.....

.....

⇒ `addl $8,%esp`  
`movl %eax,%eax`  
`movl %eax,-12(%ebp)`

.L1:

`movl %ebp,%esp`  
`popl %ebp`  
`ret`



## 스택을 이용한 변수 할당 (5)

main:

.....  
.....

addl \$8,%esp

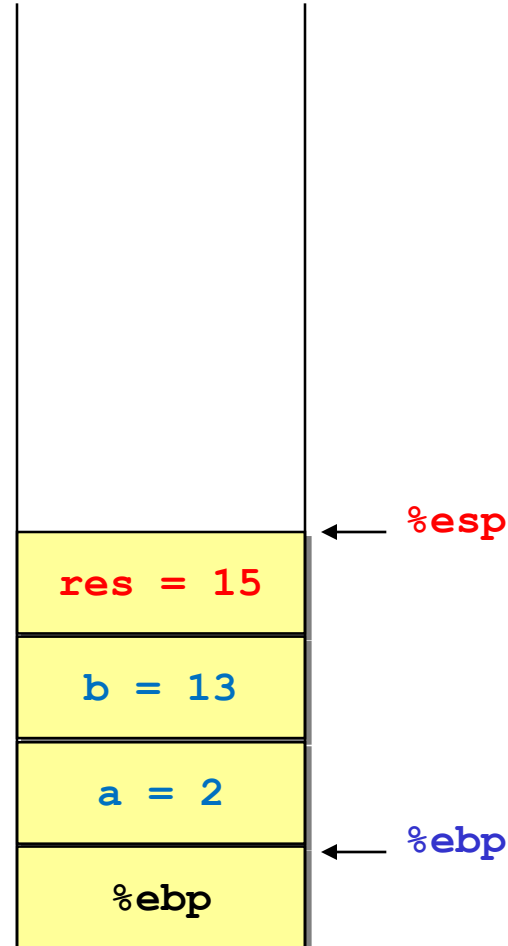
⇒ {  
  movl %eax,%eax  
  movl %eax,-12(%ebp)

.L1:

movl %ebp,%esp

popl %ebp

ret



## 스택을 이용한 변수 할당 (5)

main:

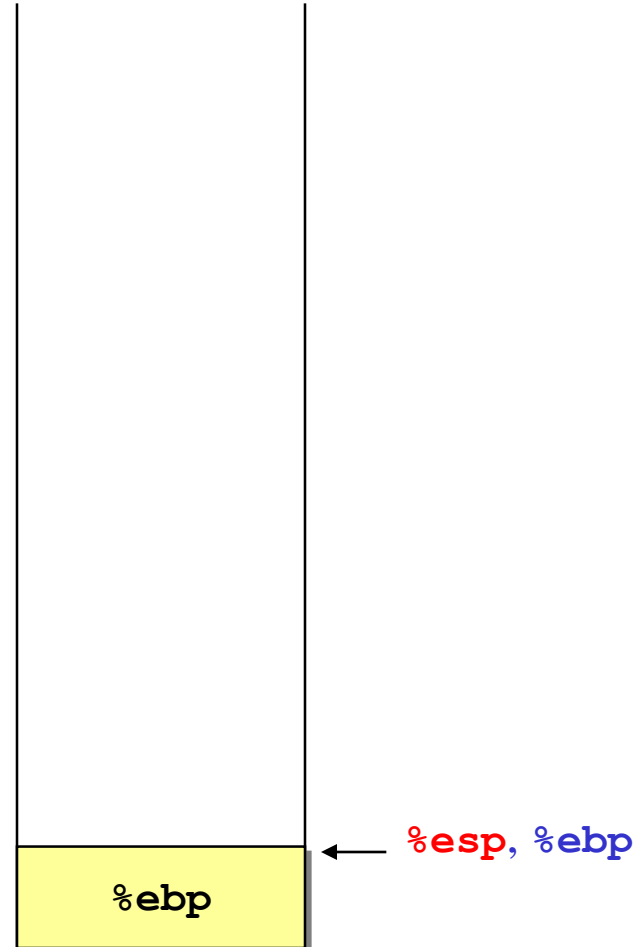
.....

.....

```
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```

.L1:

```
⇒ movl %ebp,%esp
   popl %ebp
   ret
```



## 스택을 이용한 변수 할당 (5)

main:

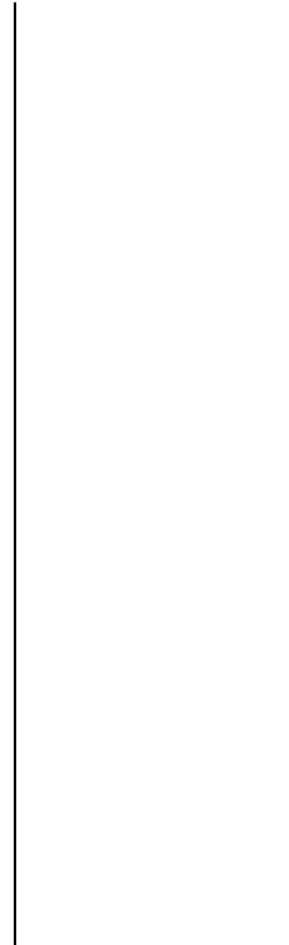
.....

.....

```
addl $8,%esp
movl %eax,%eax
movl %eax,-12(%ebp)
```

.L1:

```
movl %ebp,%esp
➡ popl %ebp
ret
```



← %esp Seoul National University