Pointers in Assembly

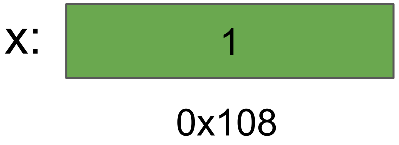
Adalbert **Gerald** Soosai Raj

swap() in C

```c
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```

```c
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```

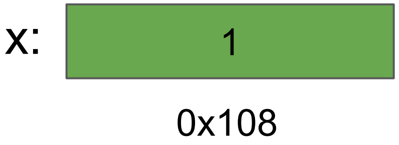
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



x: 1

0x108

```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```

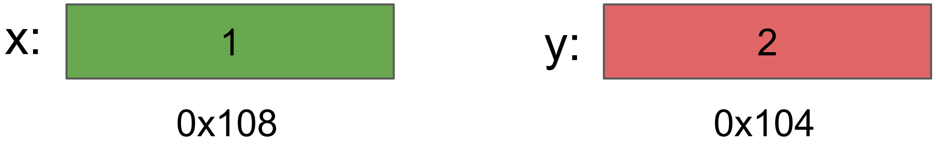


x:

1

0x108

```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



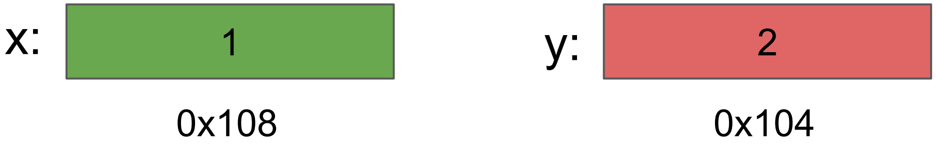
x: 1

0x108

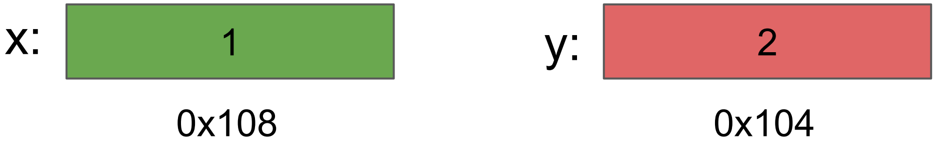
y: 2

0x104

```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



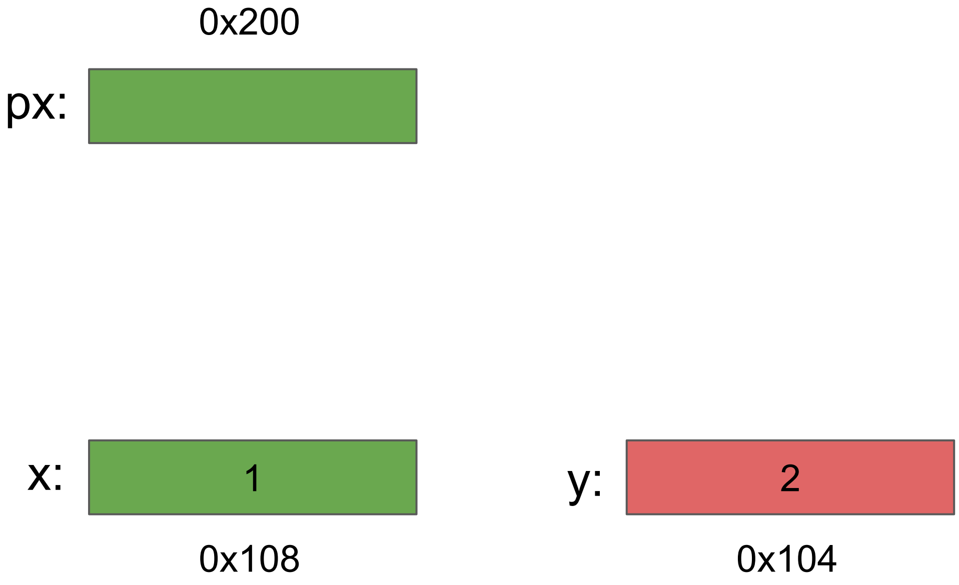
x: 1

0x108

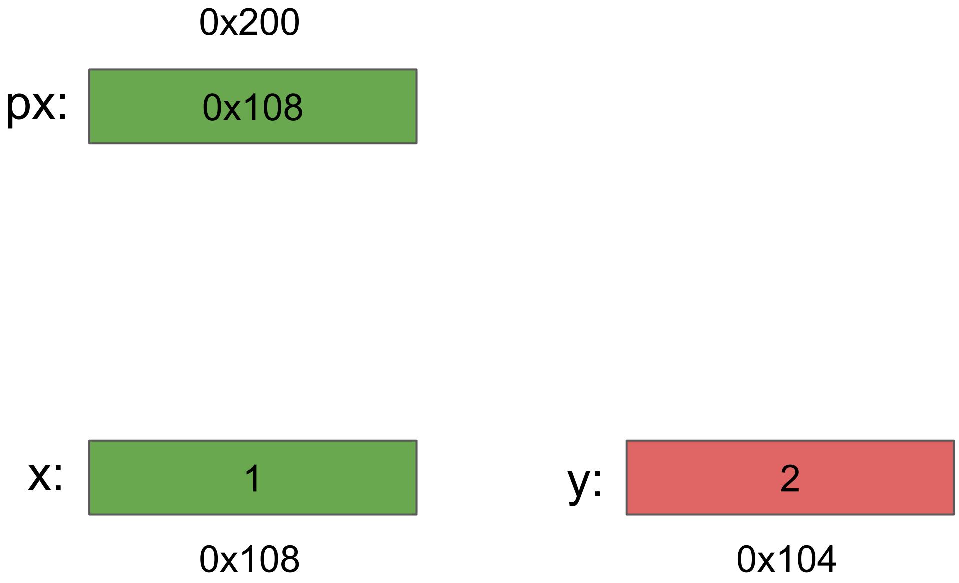
y: 2

0x104

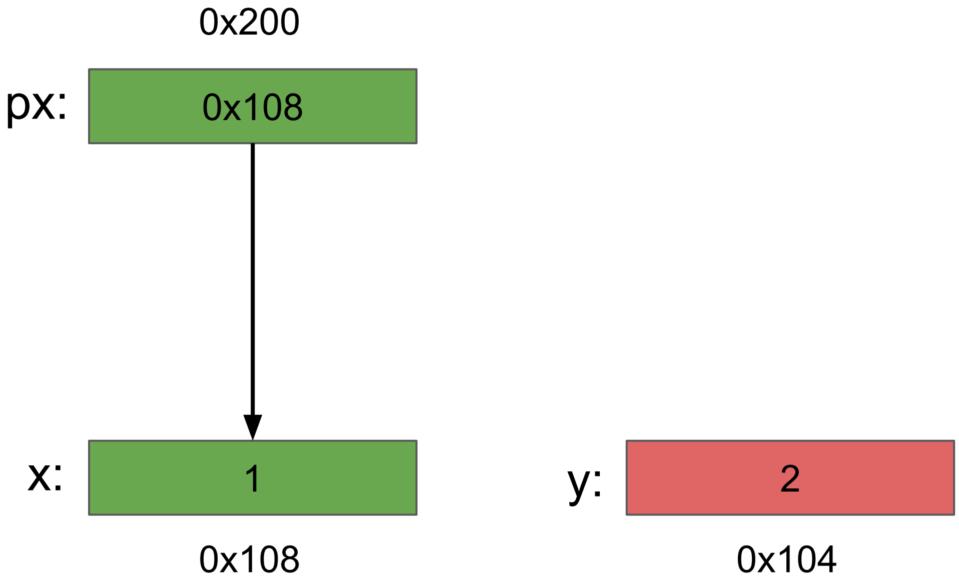
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



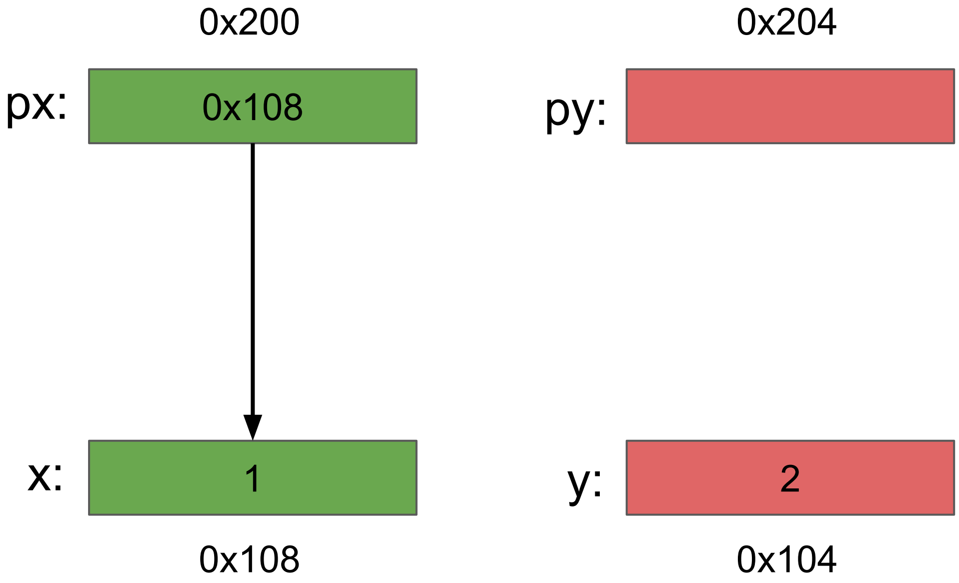
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



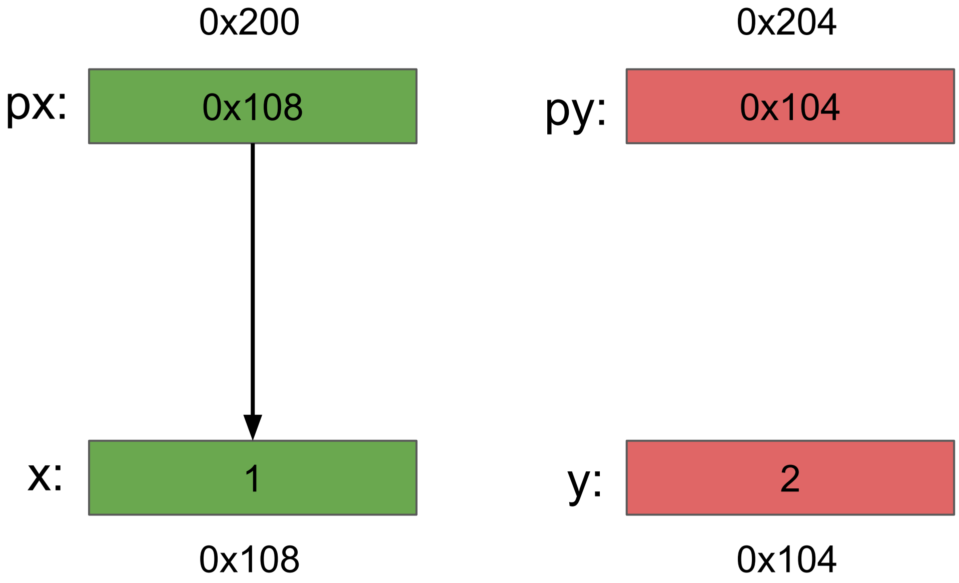
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



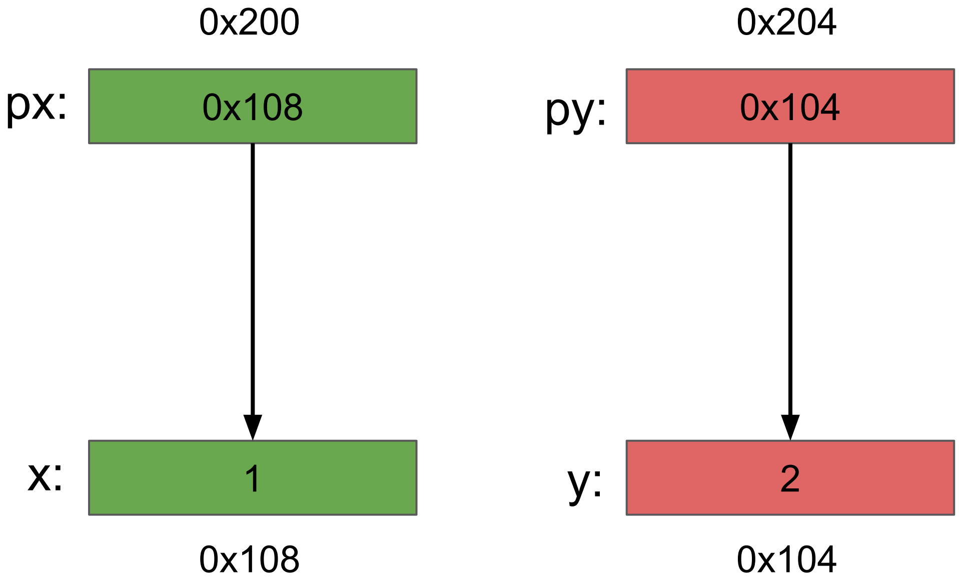
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



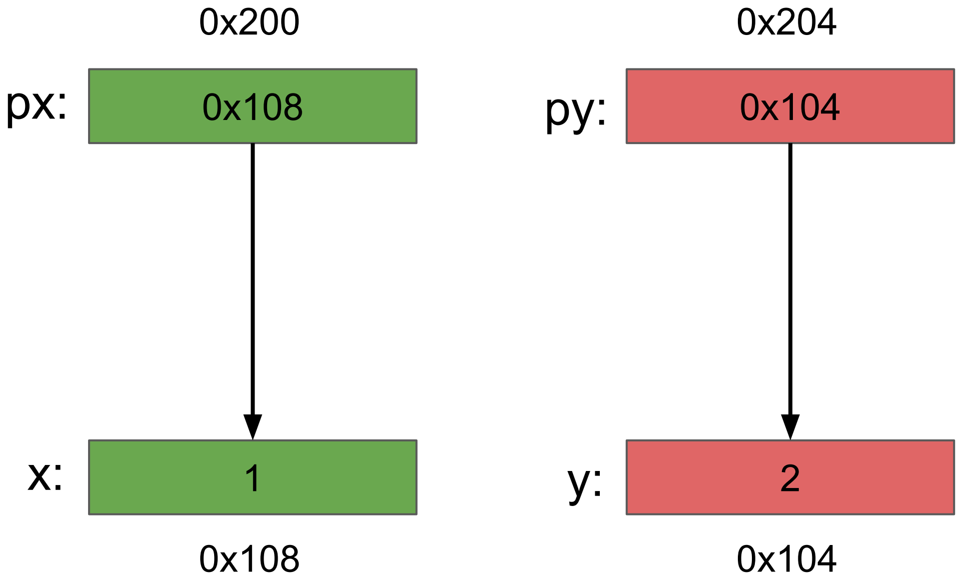
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



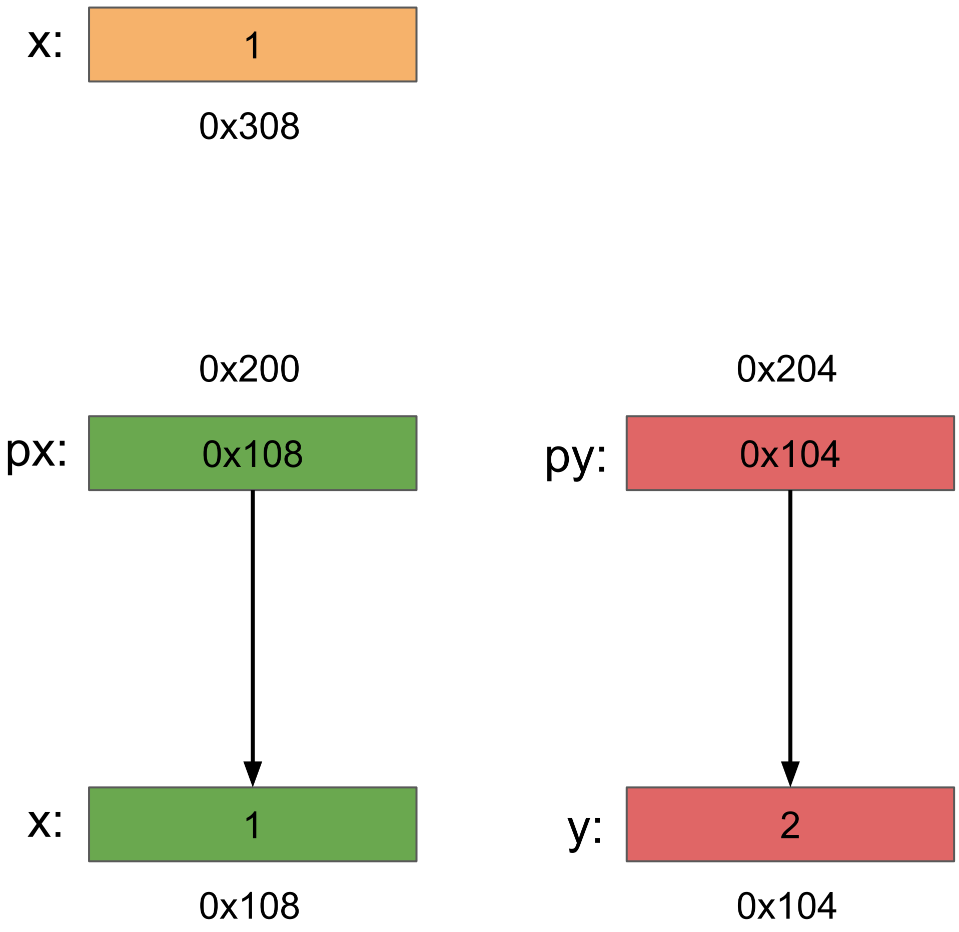
```c
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



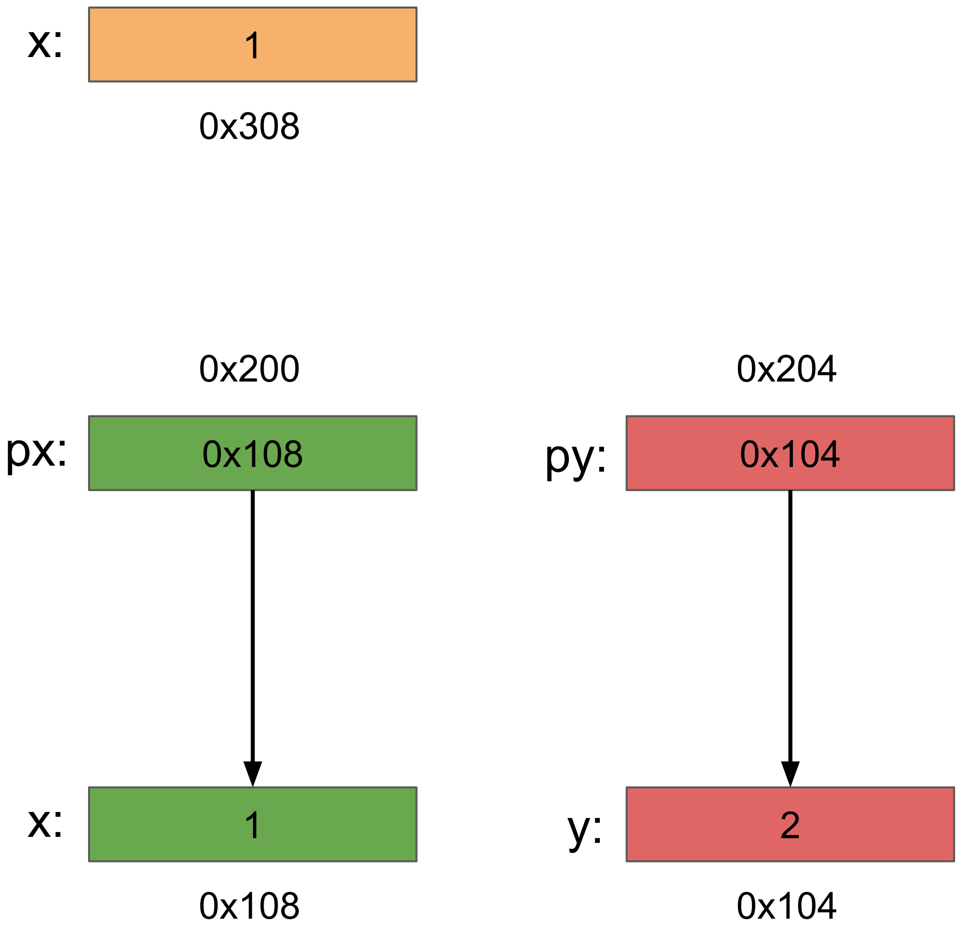
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



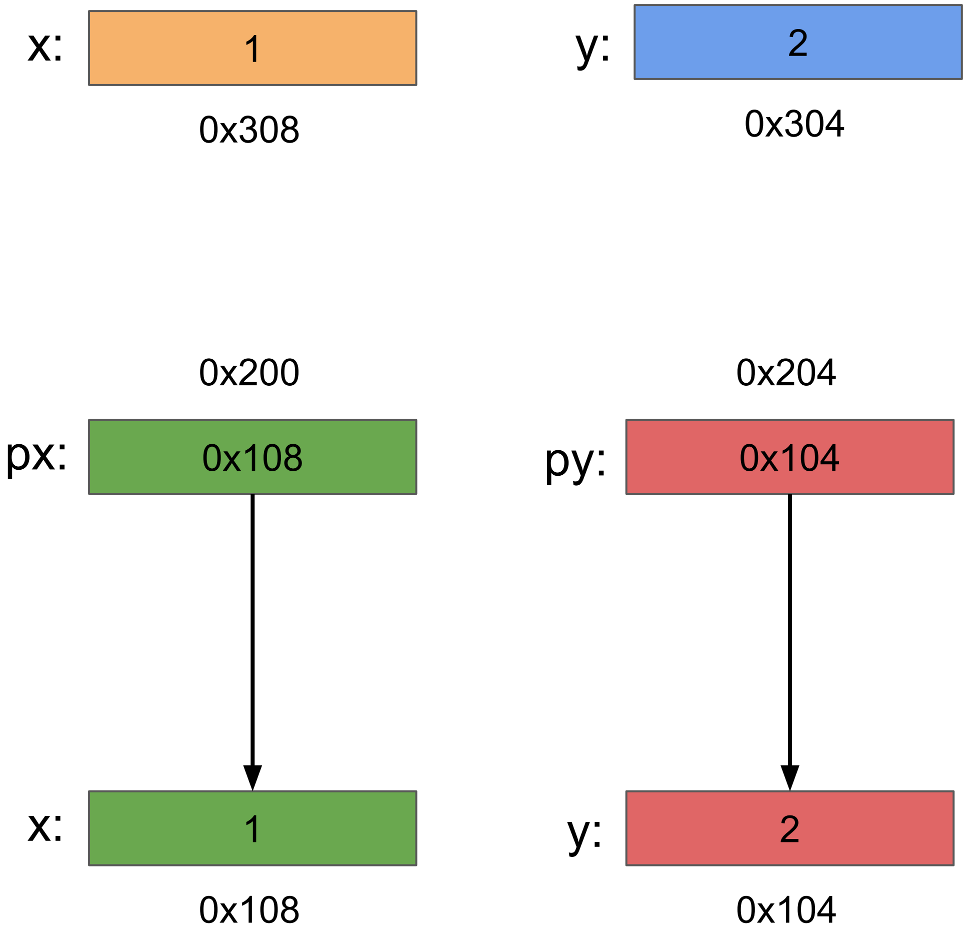
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



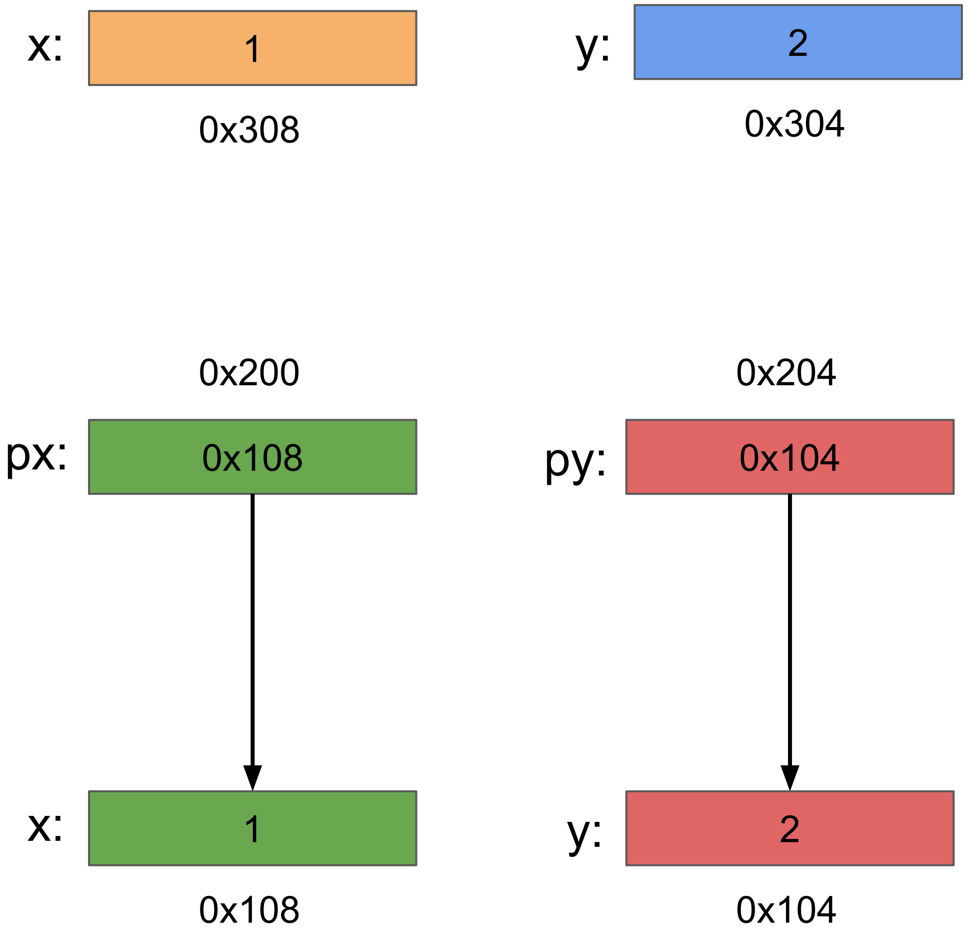
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



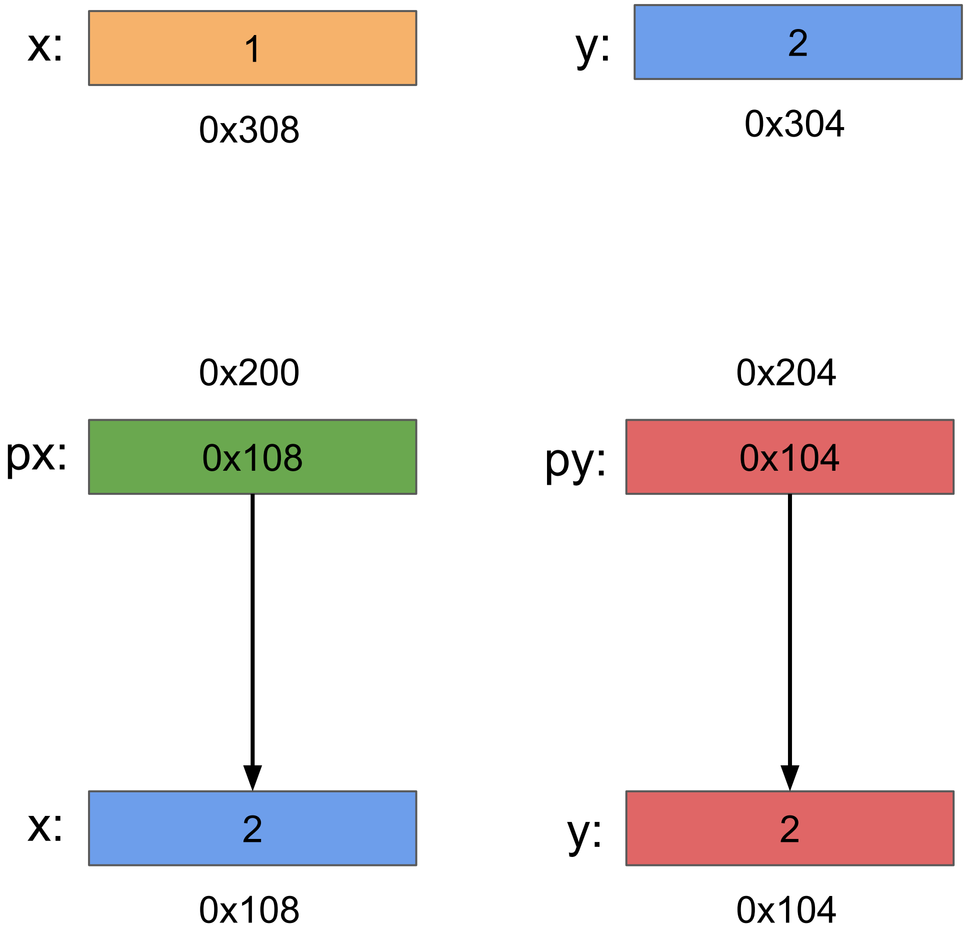
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



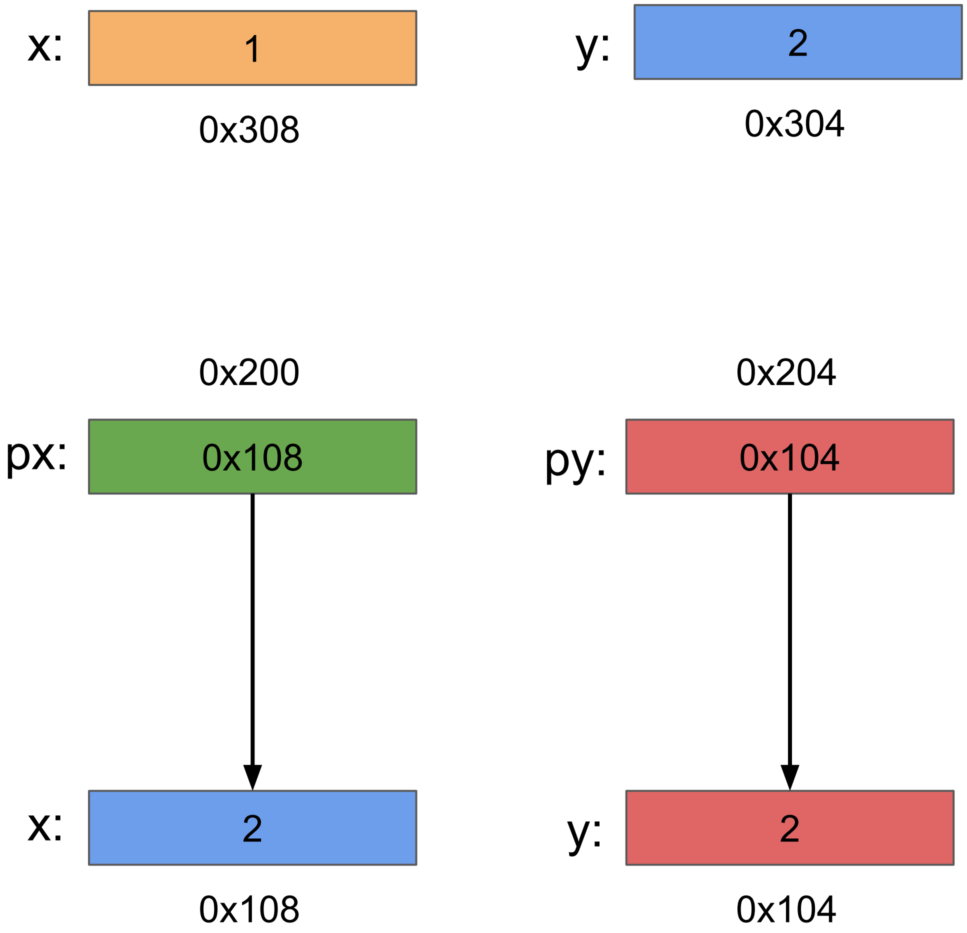
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



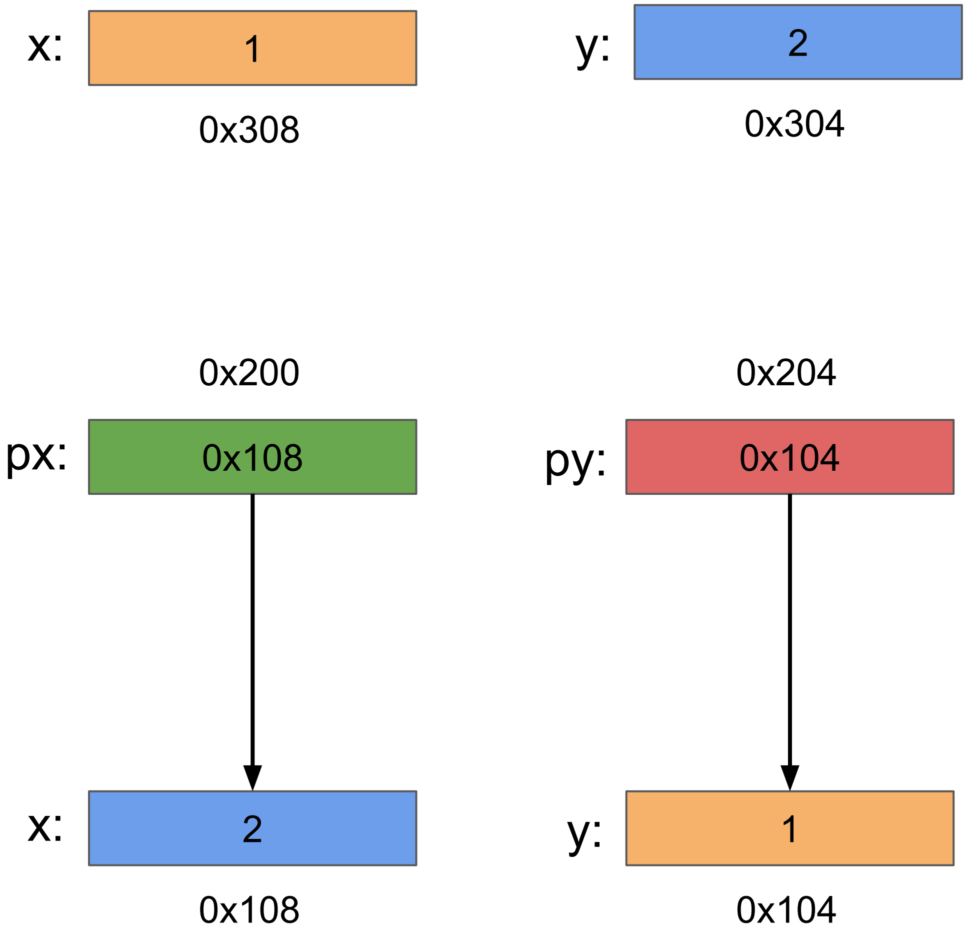
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



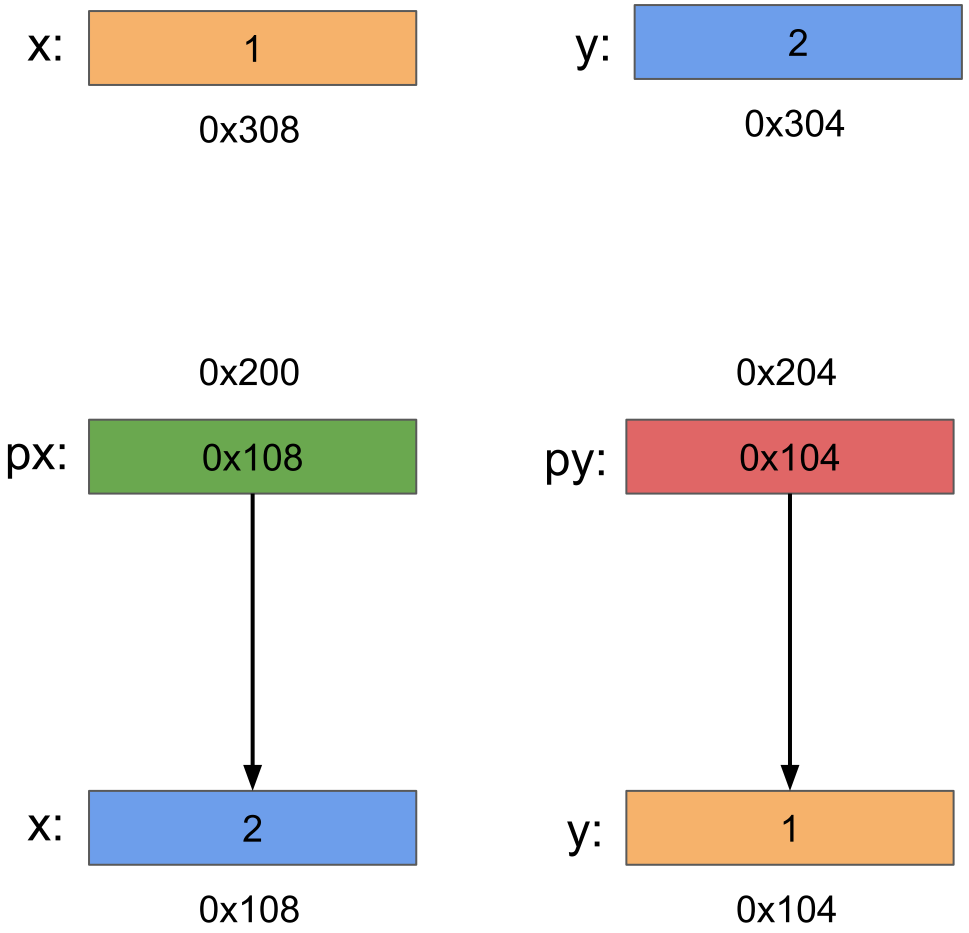
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



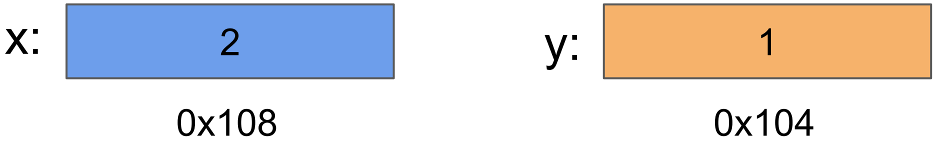
```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```

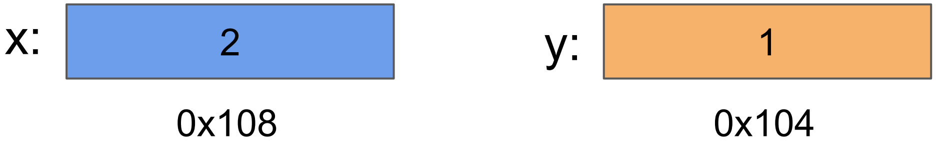


```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```



```
void swap(int *px, int *py)
{
    int x = *px;
    int y = *py;

    *px = y;
    *py = x;
}

int main()
{
    int x = 1;
    int y = 2;
    swap(&x, &y);
}
```

Control returns to main!



x: 2

0x108

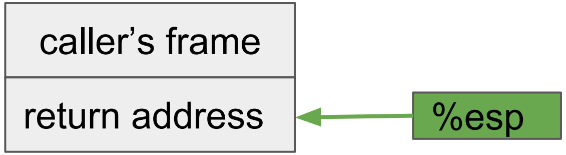
y: 1

0x104

Assembly

main() function

```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

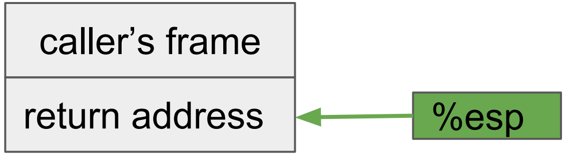


caller’s frame

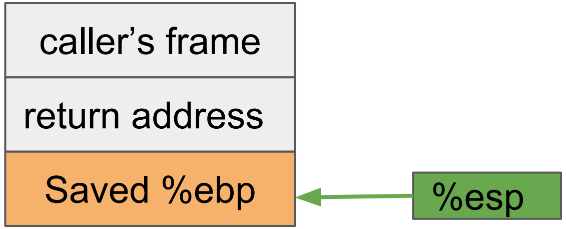
return address

%esp

```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



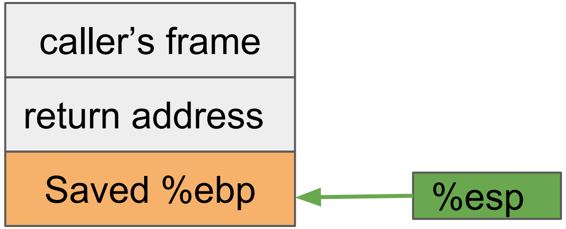
caller’s frame

return address

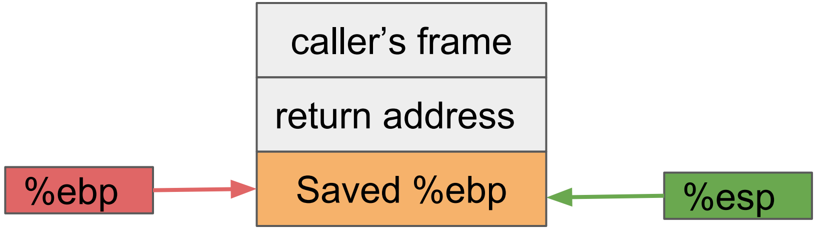
Saved %ebp

%esp

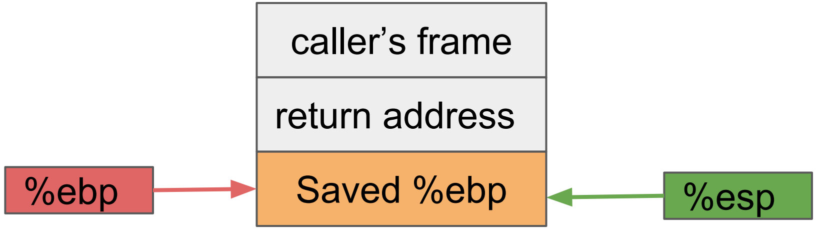
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



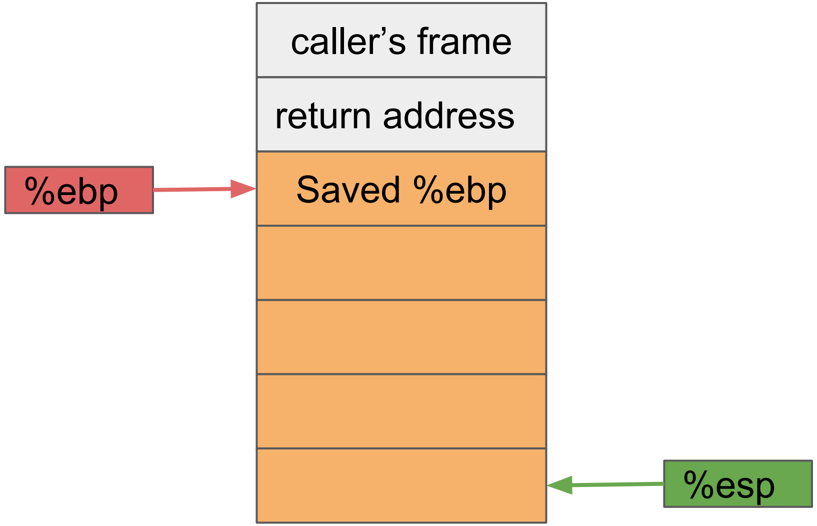
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



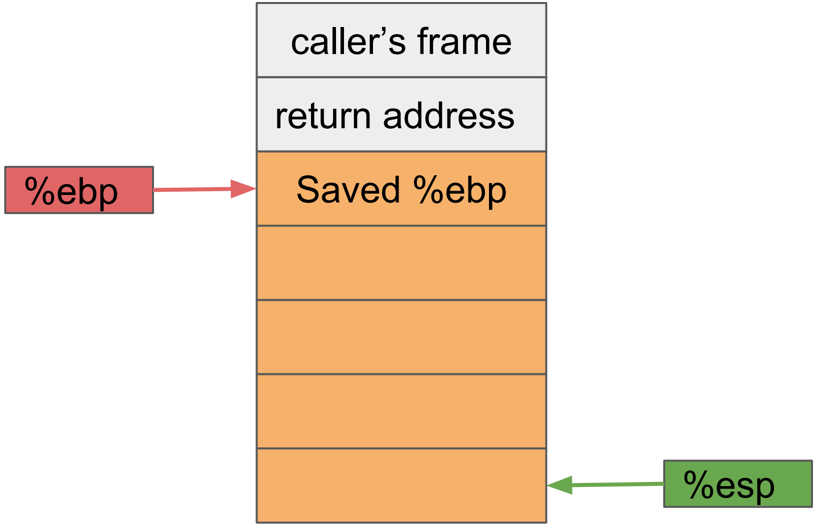
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



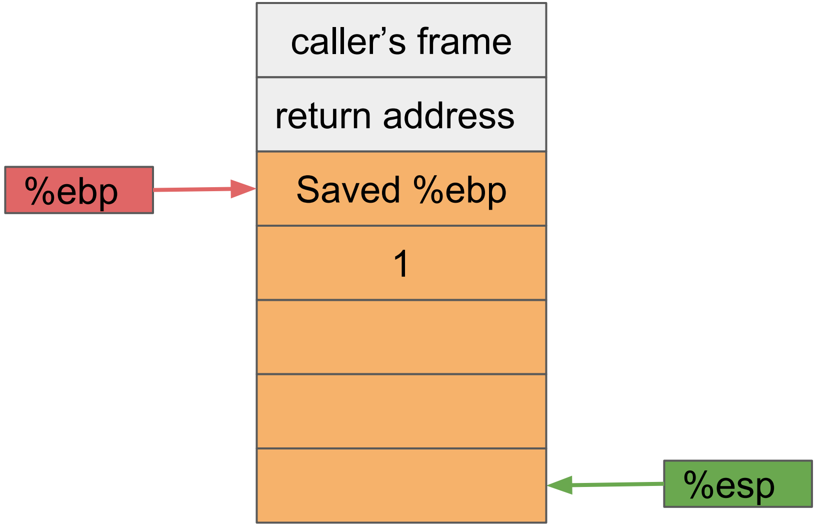
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



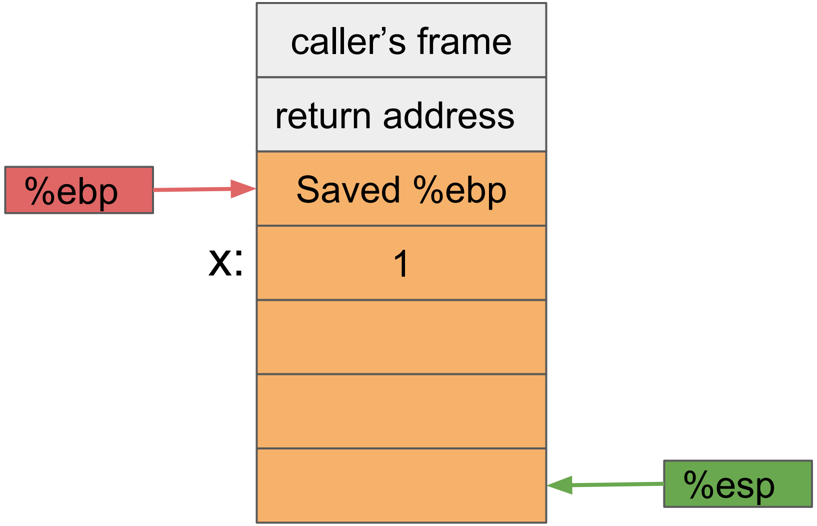
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



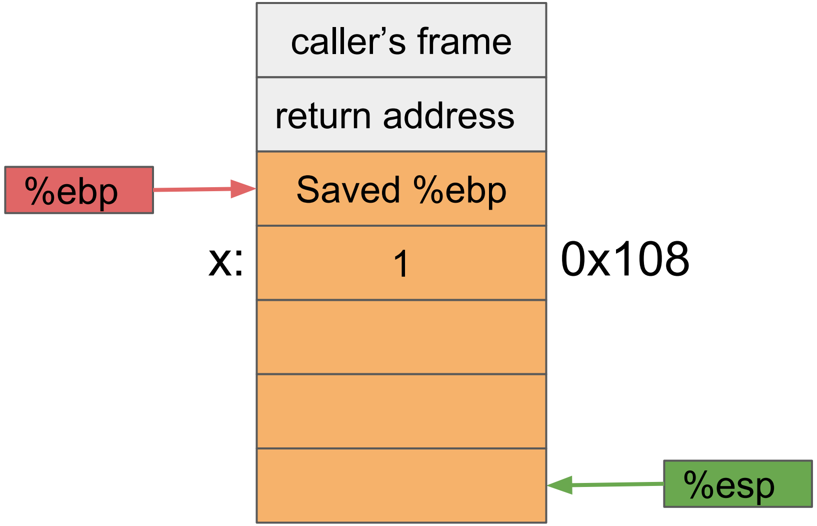
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



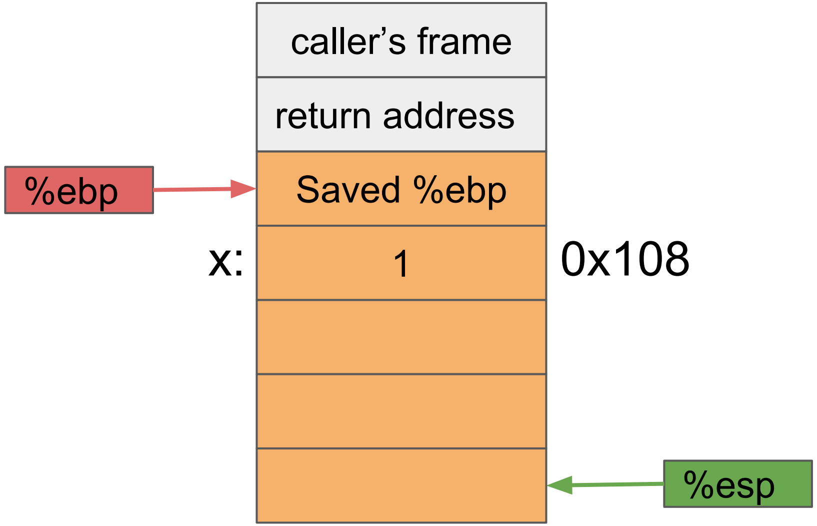
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



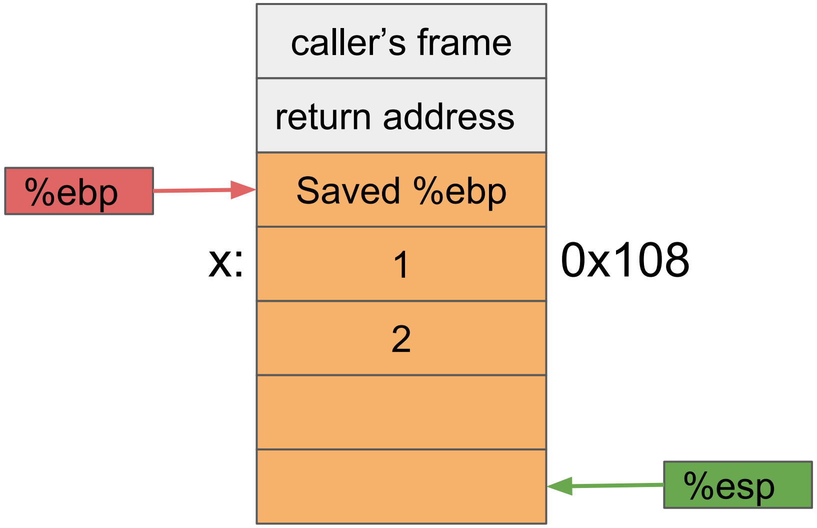
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



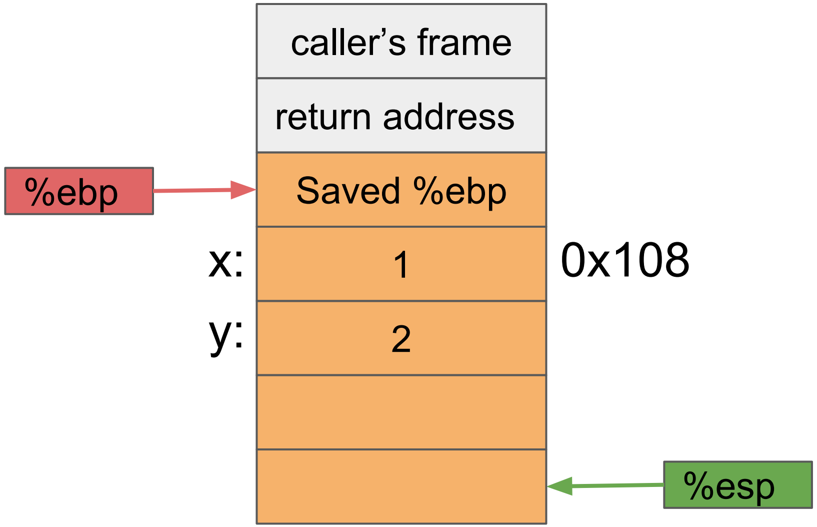
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



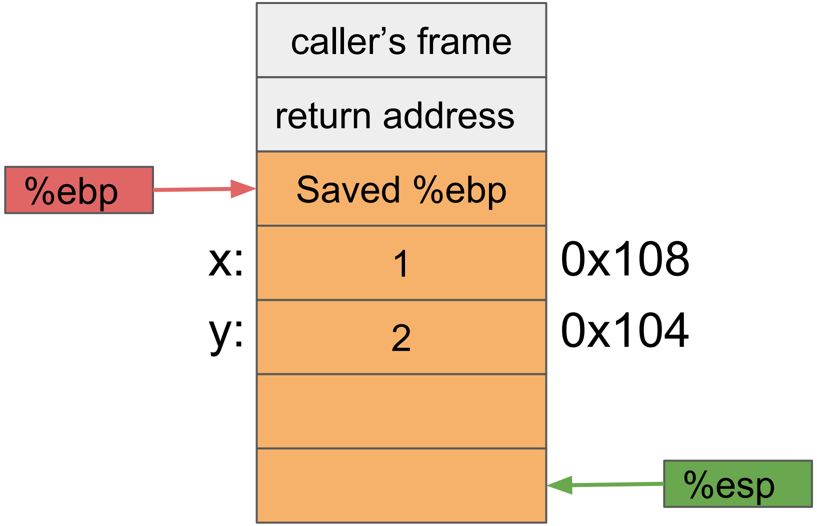
```
main:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $16, %esp
    movl     $1, -4(%ebp)
    movl     $2, -8(%ebp)
    leal     -8(%ebp), %eax
    pushl    %eax
    leal     -4(%ebp), %eax
    pushl    %eax
    call     swap
    addl     $8, %esp
    leave
    ret
```



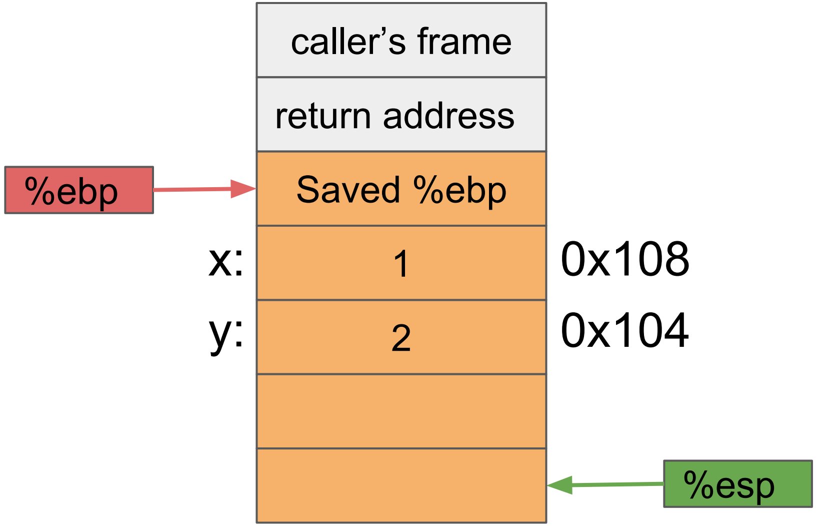
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



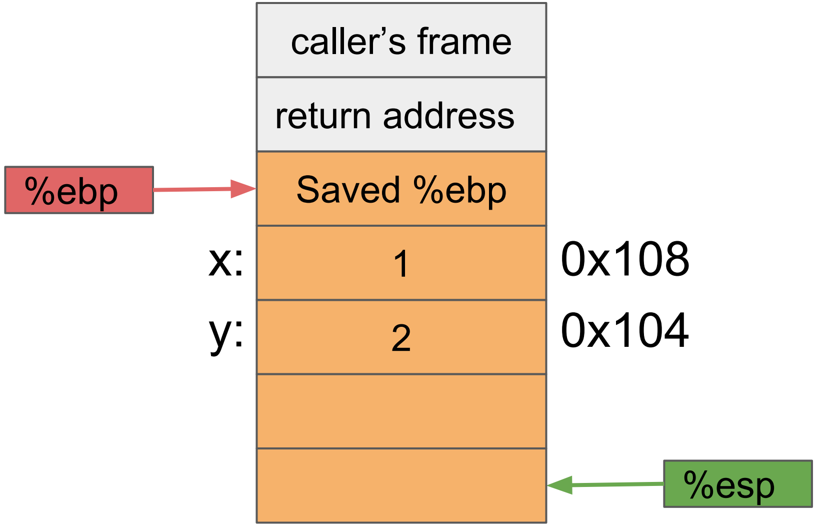
```
main:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $16, %esp
    movl     $1, -4(%ebp)
    movl     $2, -8(%ebp)
    leal     -8(%ebp), %eax
    pushl    %eax
    leal     -4(%ebp), %eax
    pushl    %eax
    call     swap
    addl     $8, %esp
    leave
    ret
```



```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

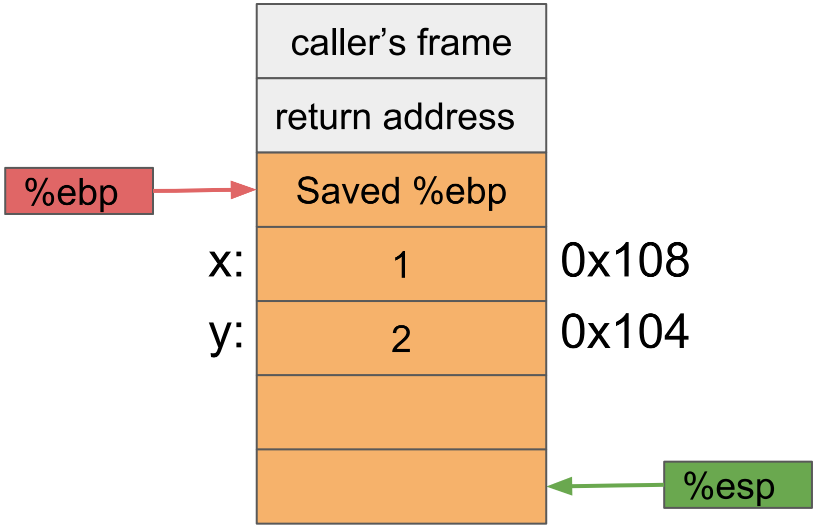


```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

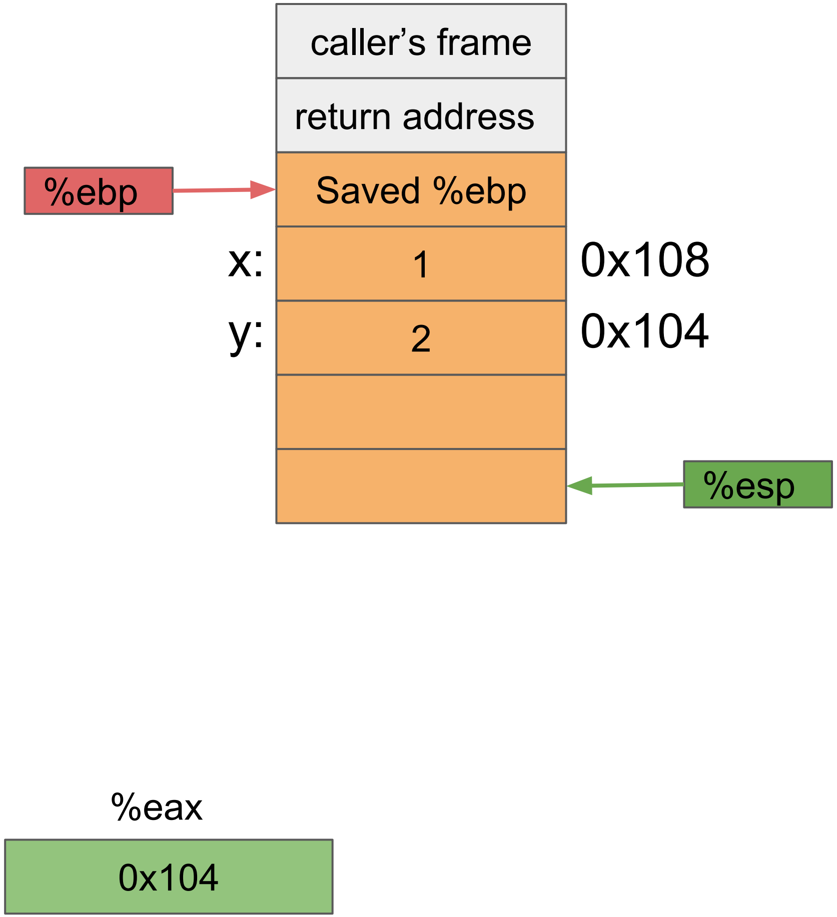


-8(%ebp) = -8 + R[%ebp] = 0x104

```
main:
     pushl    %ebp
     movl     %esp, %ebp
     subl     $16, %esp
     movl     $1, -4(%ebp)
     movl     $2, -8(%ebp)
     leal     -8(%ebp), %eax
     pushl    %eax
     leal     -4(%ebp), %eax
     pushl    %eax
     call     swap
     addl     $8, %esp
     leave
     ret
```



```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



caller’s frame

return address

%ebp → Saved %ebp

x: 1 0x108

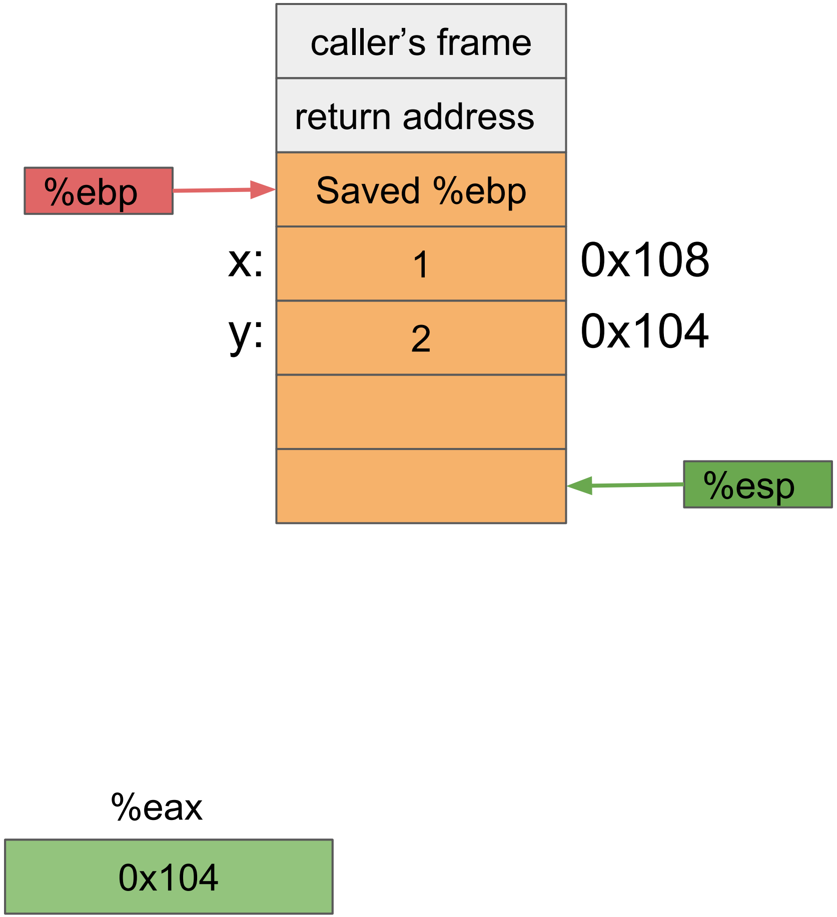
y: 2 0x104

%esp

%eax

0x104

```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



caller’s frame

return address

%ebp

Saved %ebp

x: 1 0x108

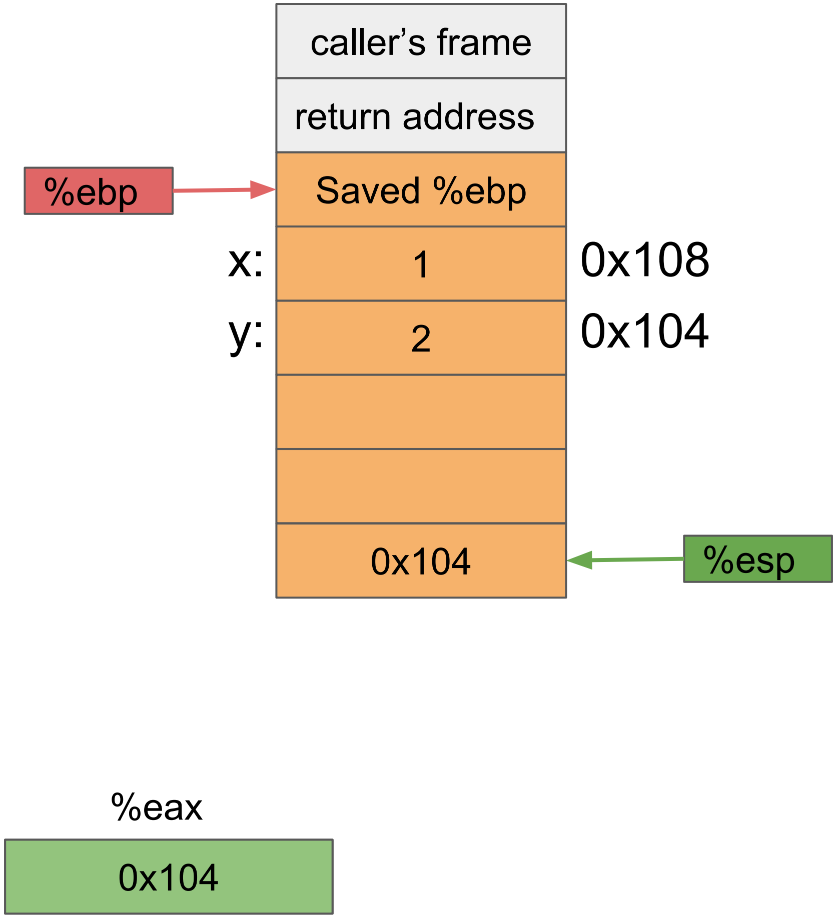
y: 2 0x104

%esp

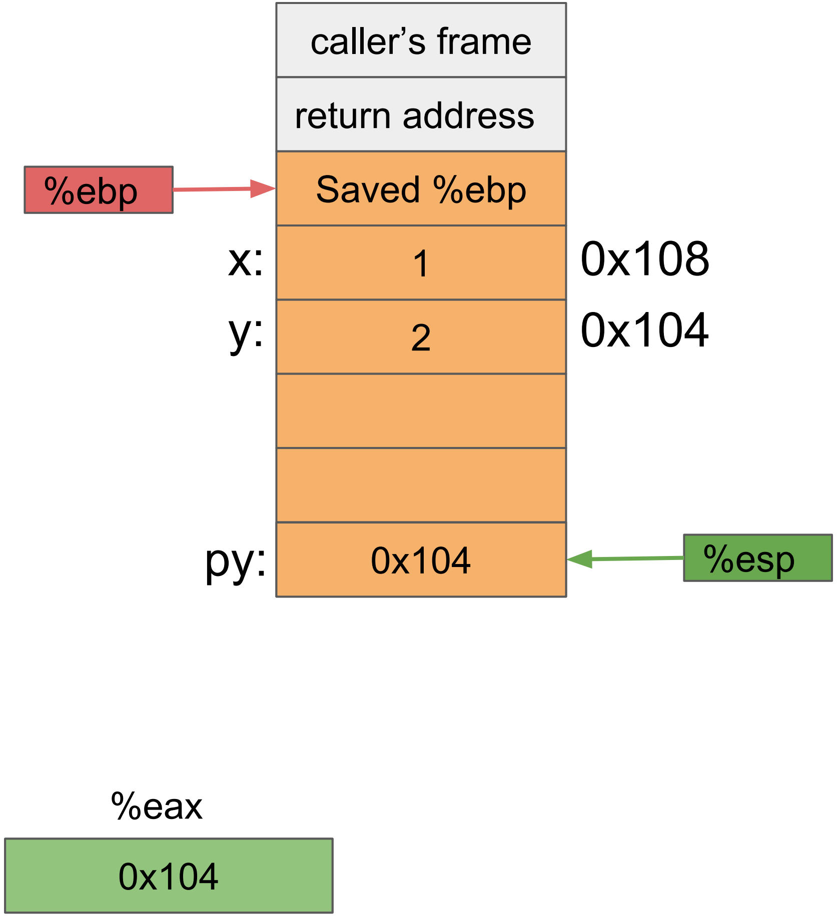
%eax

0x104

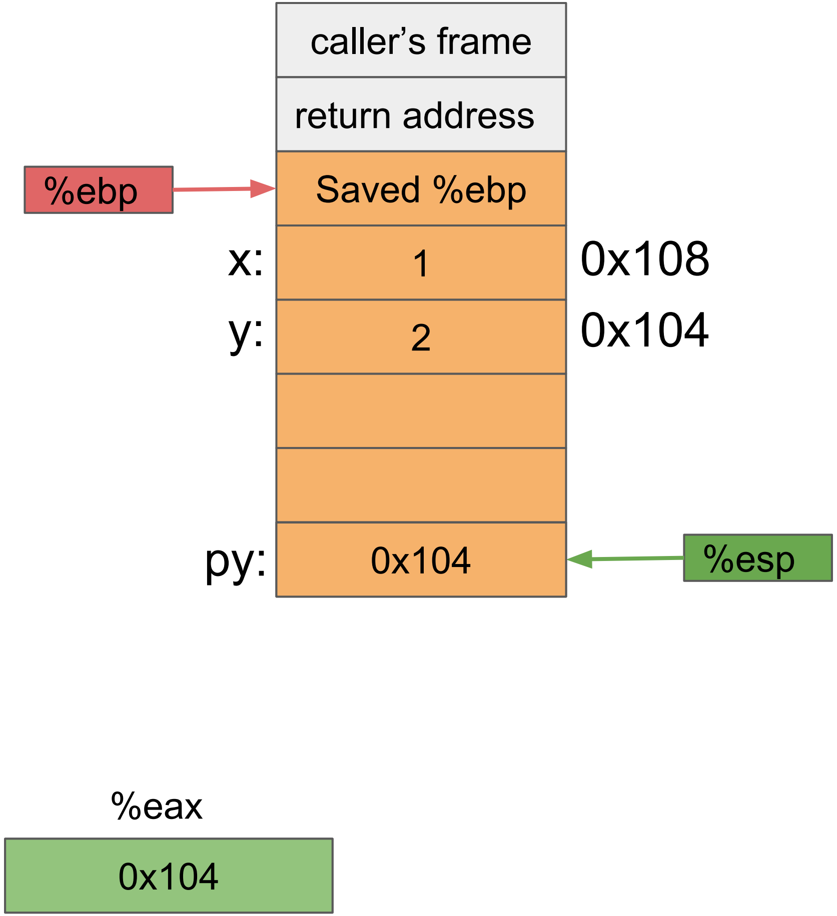
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



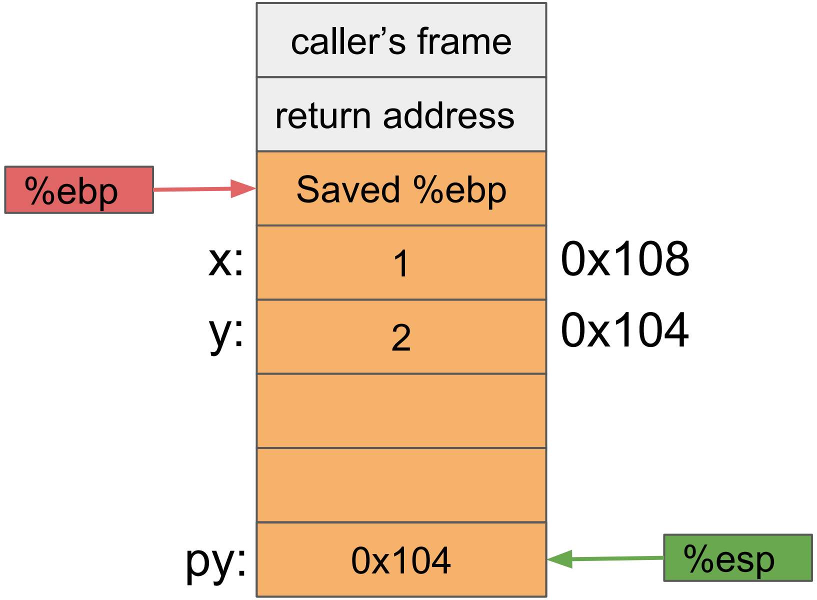
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

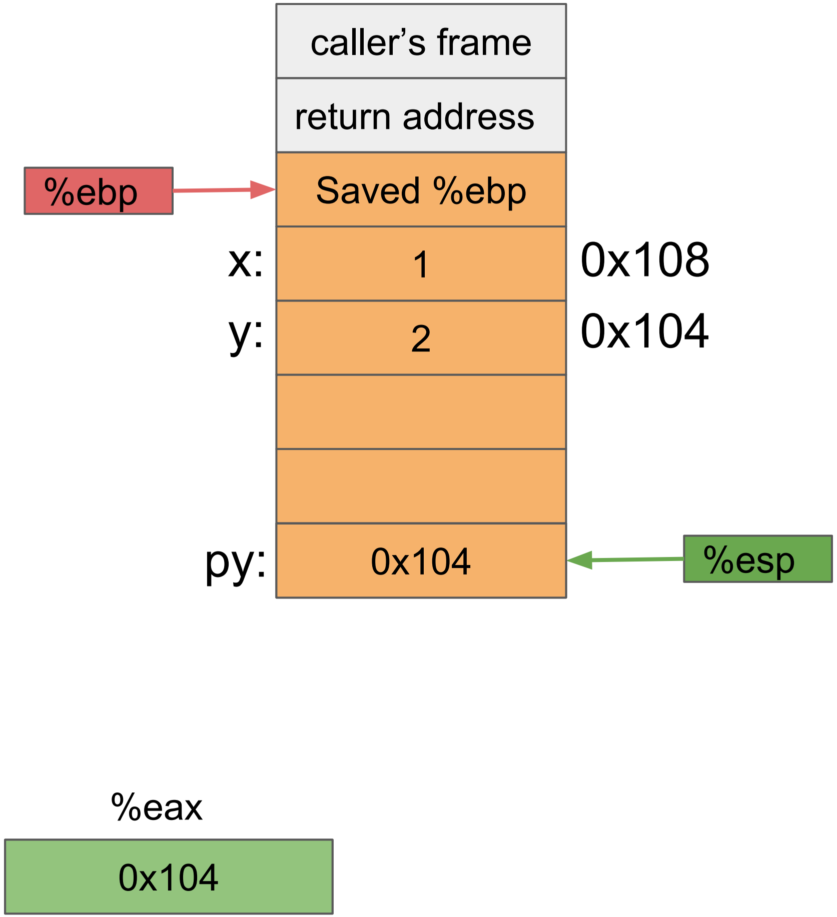


```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

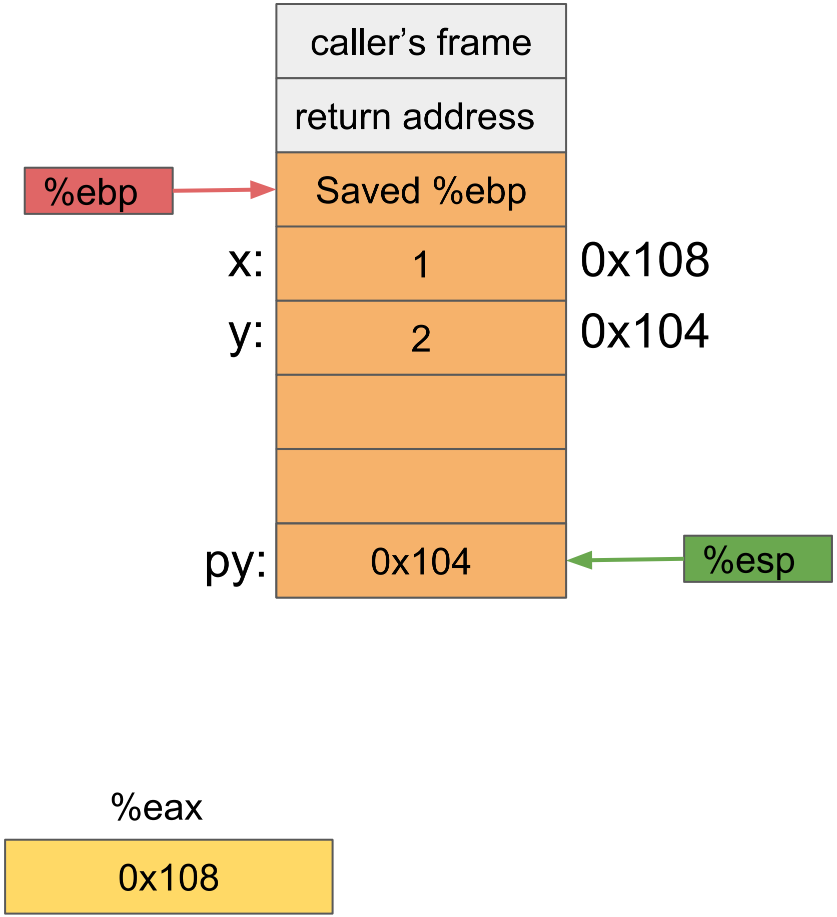


`-4(%ebp) = -8 + R[%ebp] = 0x108`

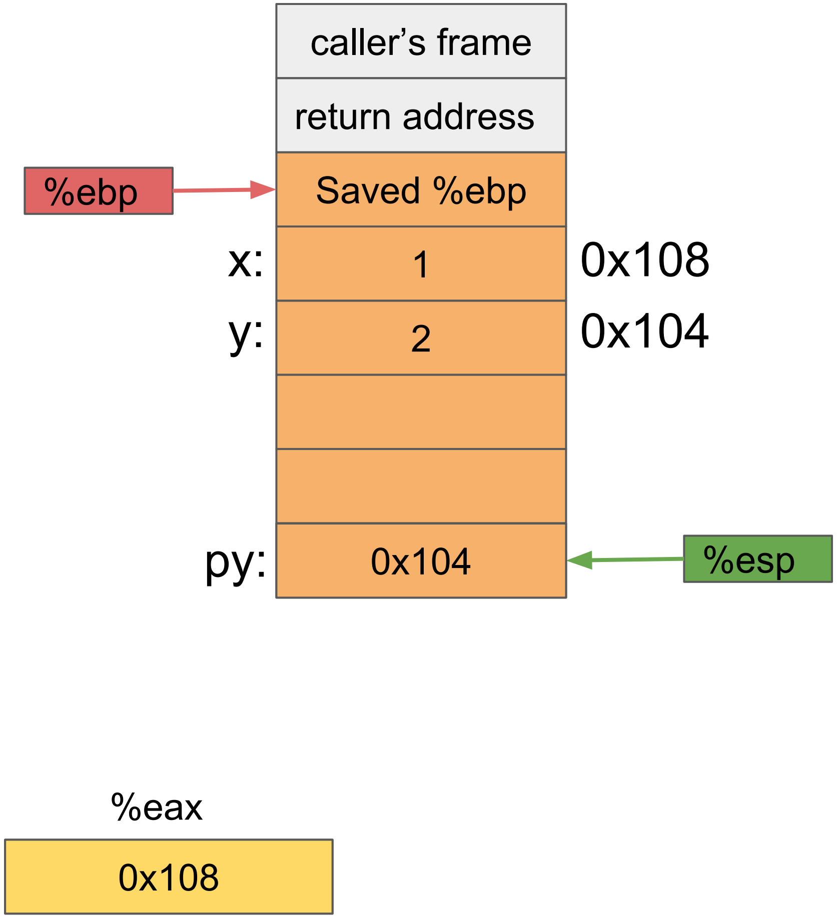
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



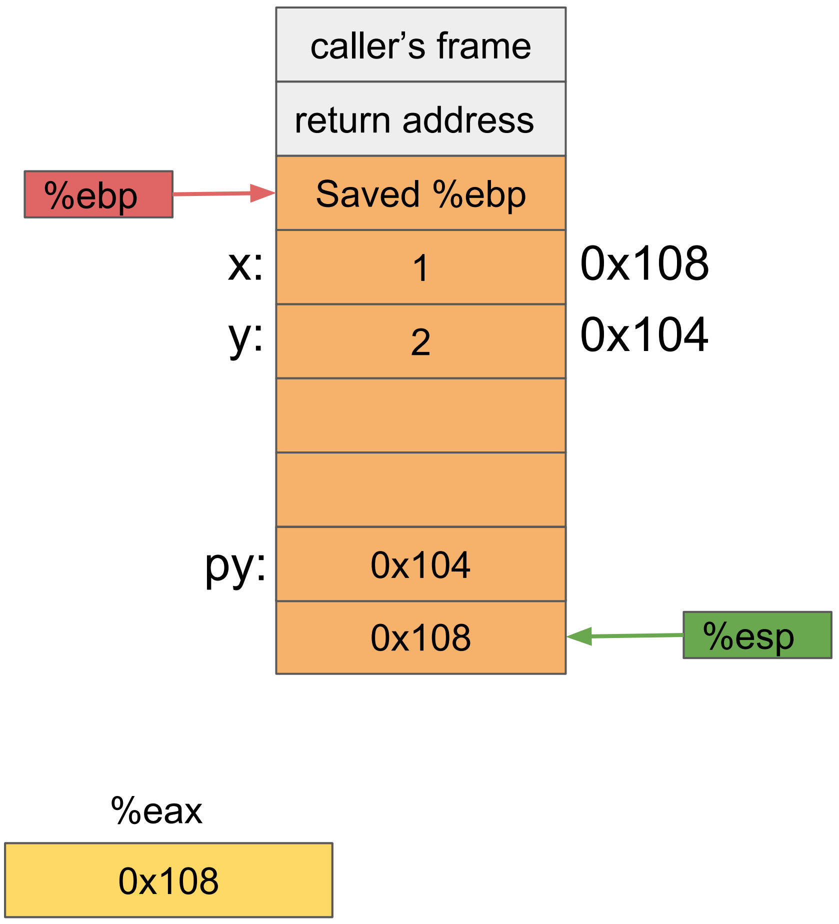
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



```
main:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $16, %esp
    movl     $1, -4(%ebp)
    movl     $2, -8(%ebp)
    leal     -8(%ebp), %eax
    pushl    %eax
    leal     -4(%ebp), %eax
    pushl    %eax
    call     swap
    addl     $8, %esp
    leave
    ret
```



caller’s frame

return address

%ebp

Saved %ebp

x: 1 0x108

y: 2 0x104

py: 0x104

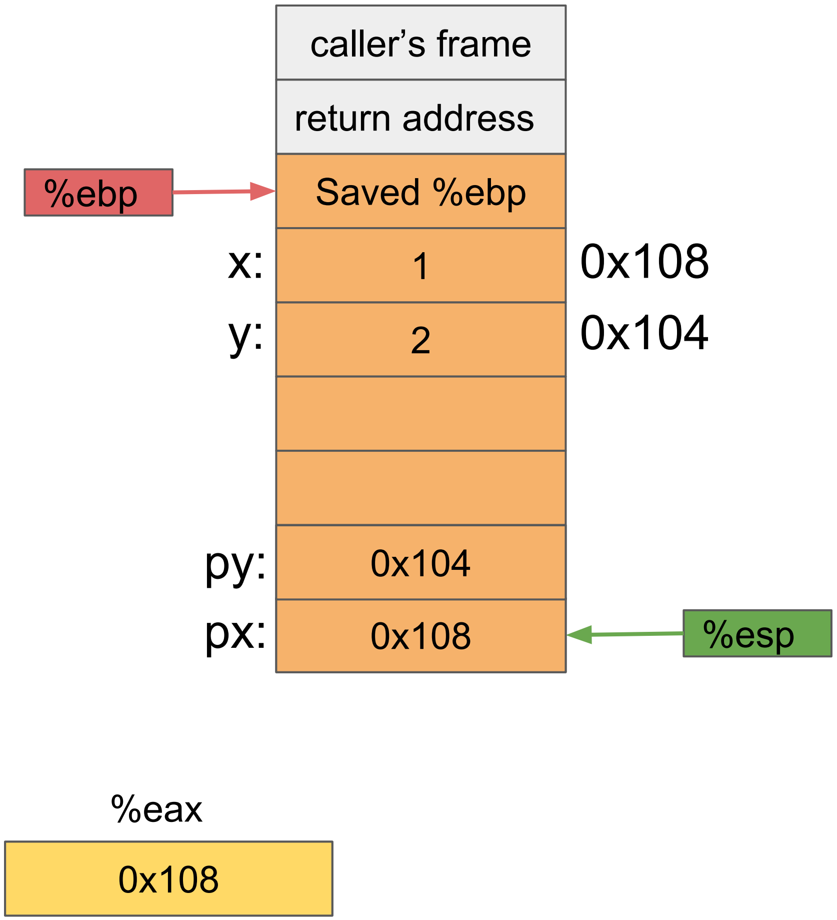
0x108

%esp

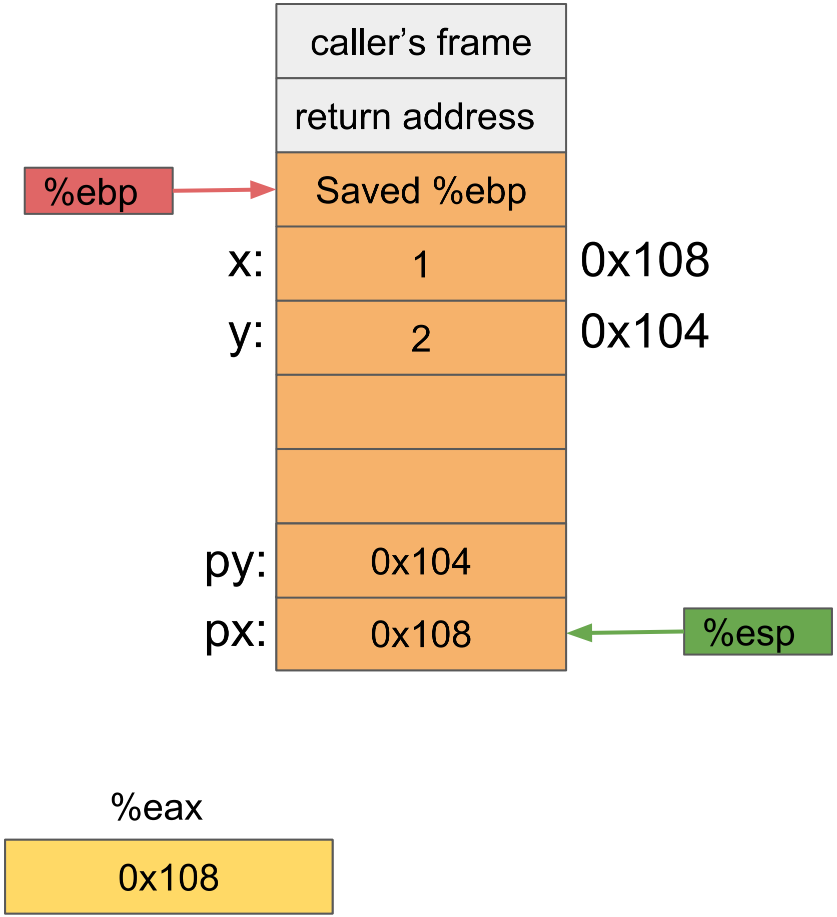
%eax

0x108

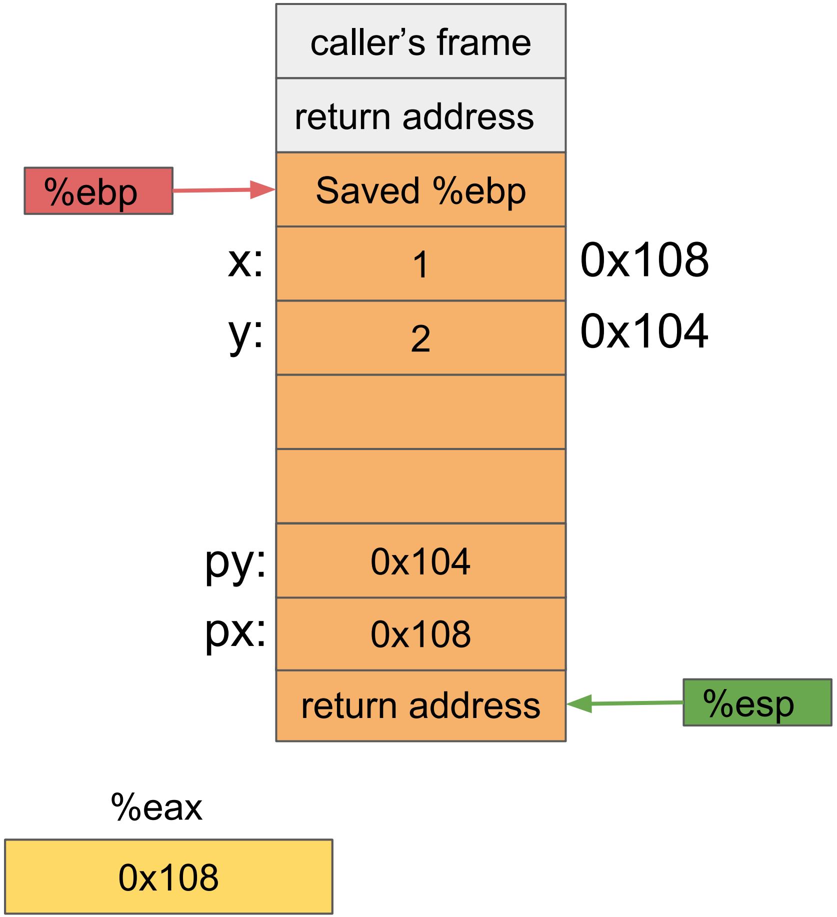
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

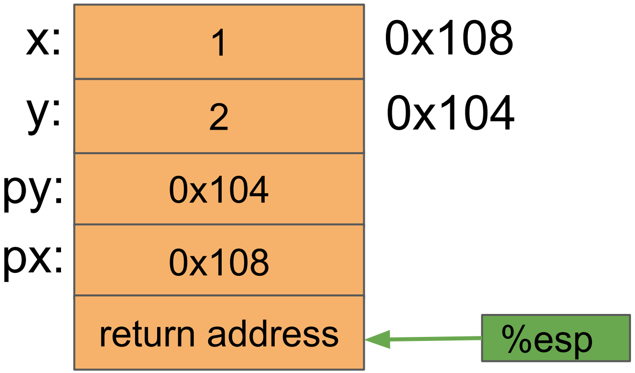


```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

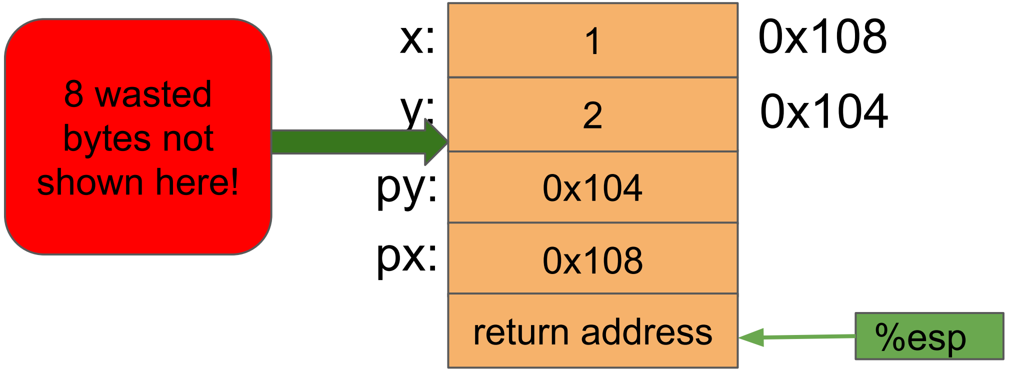


swap() function

```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```

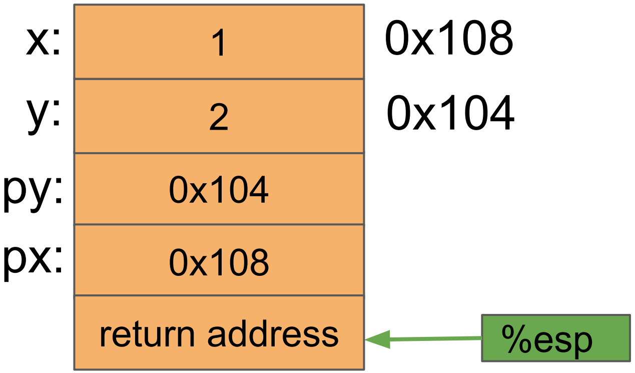


```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```

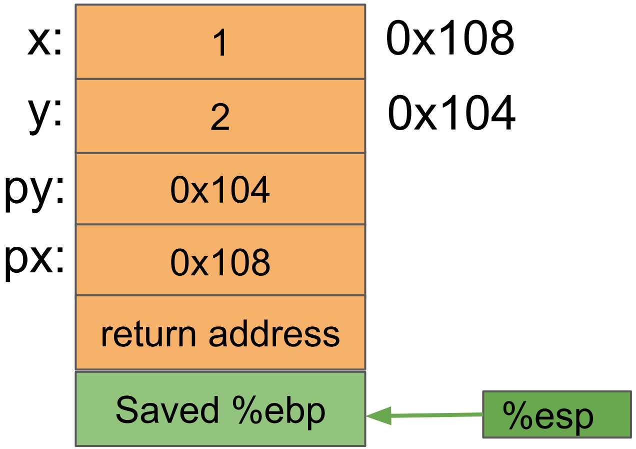


8 wasted bytes not shown here!

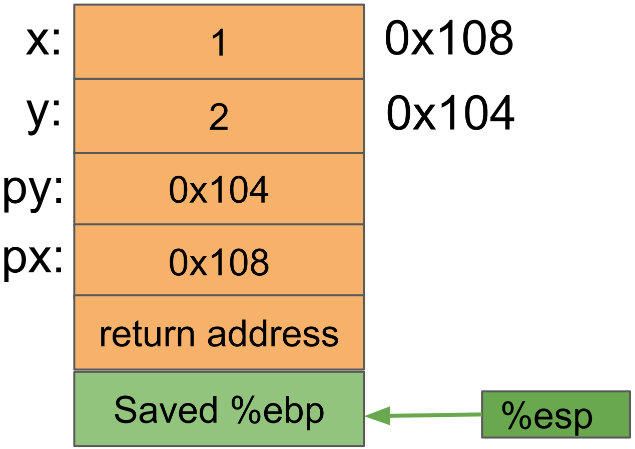
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $16, %esp
    movl     8(%ebp), %eax
    movl     (%eax), %eax
    movl     %eax, -4(%ebp)
    movl     12(%ebp), %eax
    movl     (%eax), %eax
    movl     %eax, -8(%ebp)
    movl     8(%ebp), %eax
    movl     -8(%ebp), %edx
    movl     %edx, (%eax)
    movl     12(%ebp), %eax
    movl     -4(%ebp), %edx
    movl     %edx, (%eax)
    leave
    ret
```

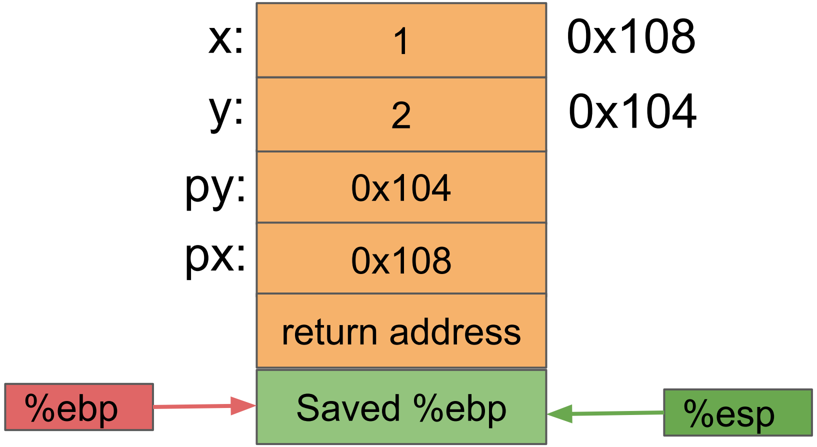


```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```

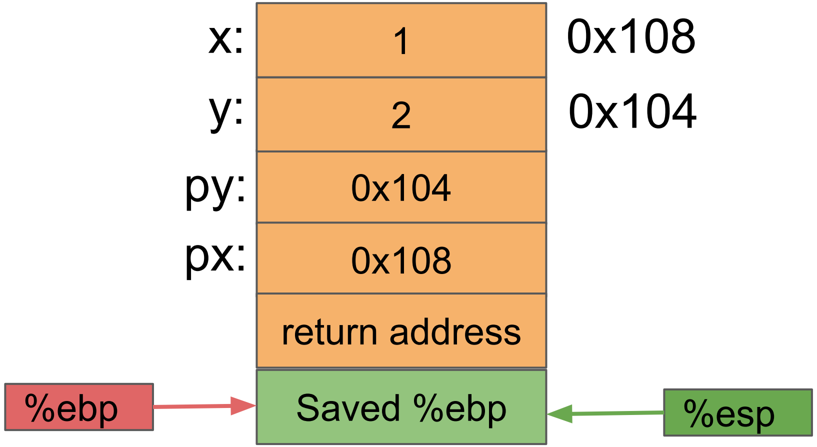


x:	1	0x108
y:	2	0x104
py:	0x104	
px:	0x108	
	return address	
	Saved %ebp	← %esp

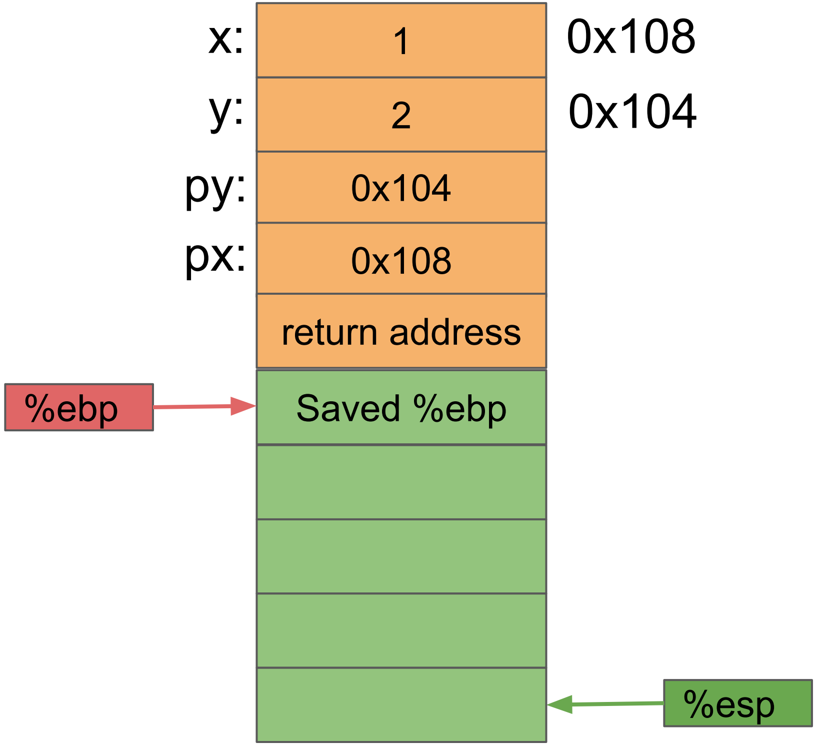
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



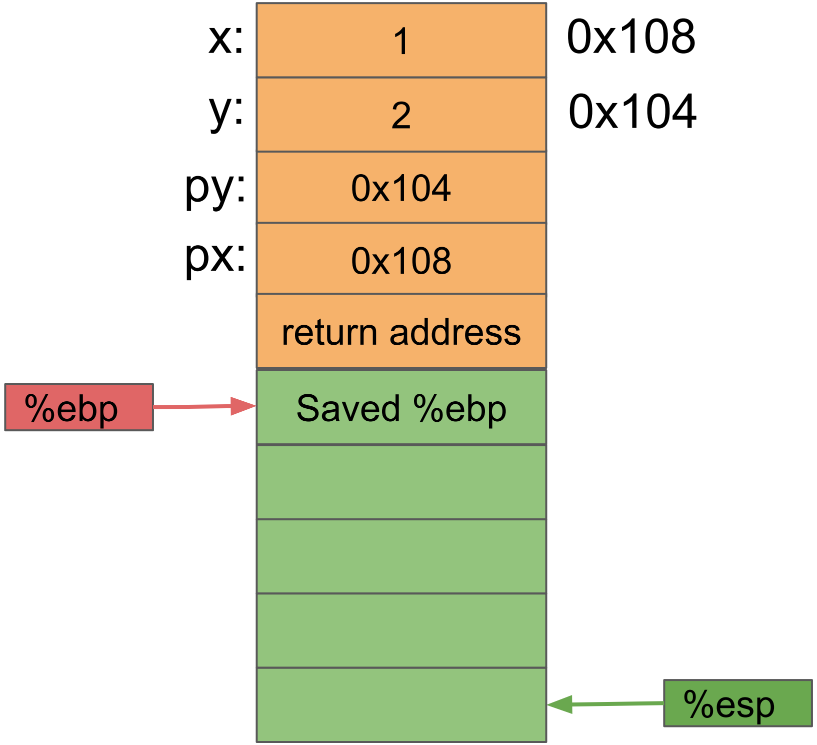
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



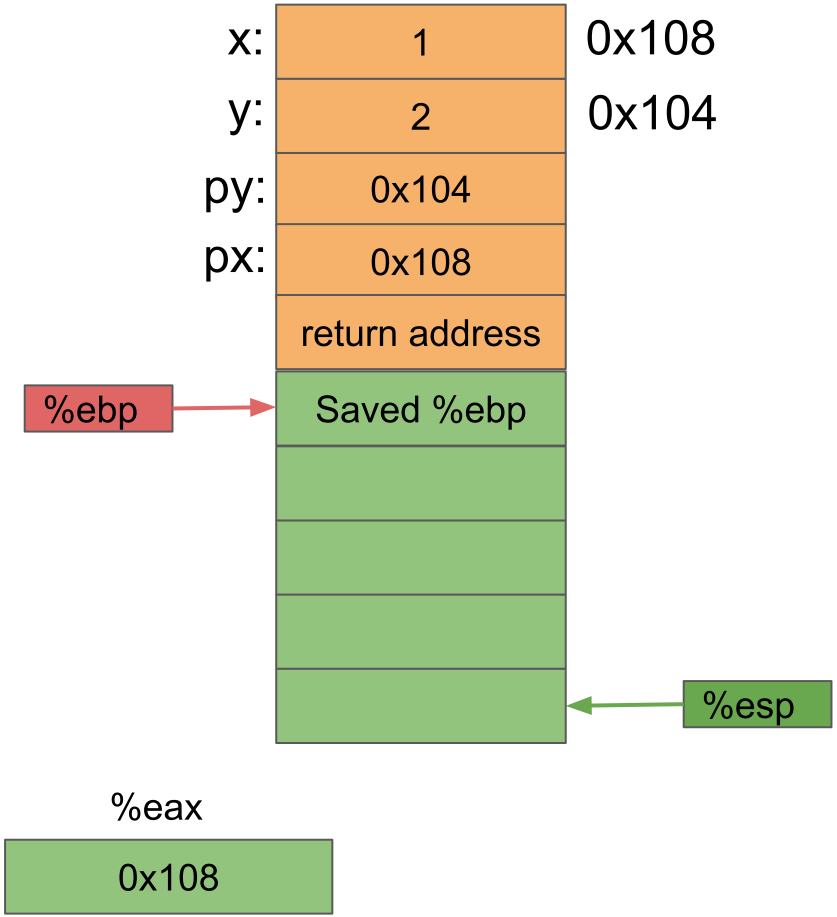
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



x: 1 0x108

y: 2 0x104

py: 0x104

px: 0x108

return address

%ebp

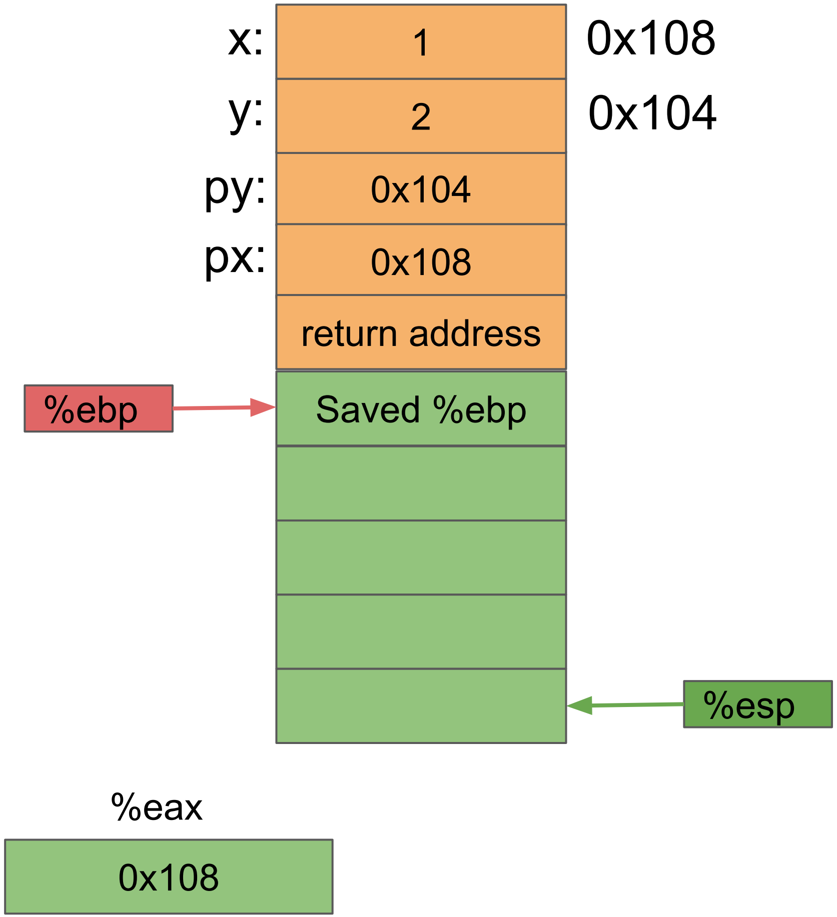
Saved %ebp

%esp

%eax

0x108

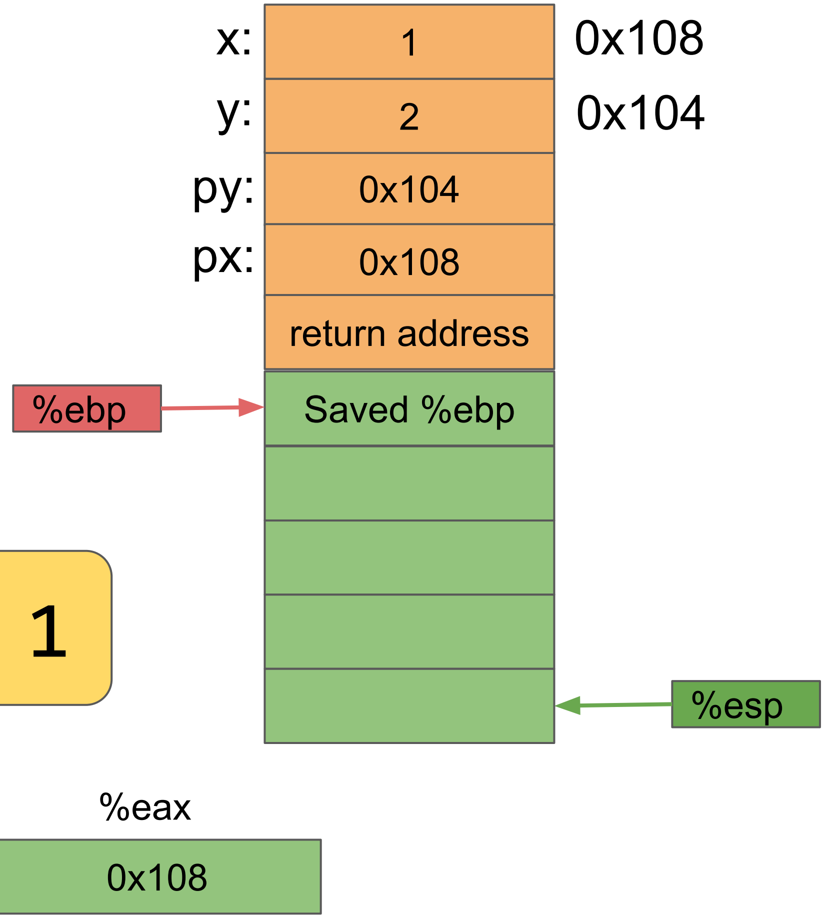
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



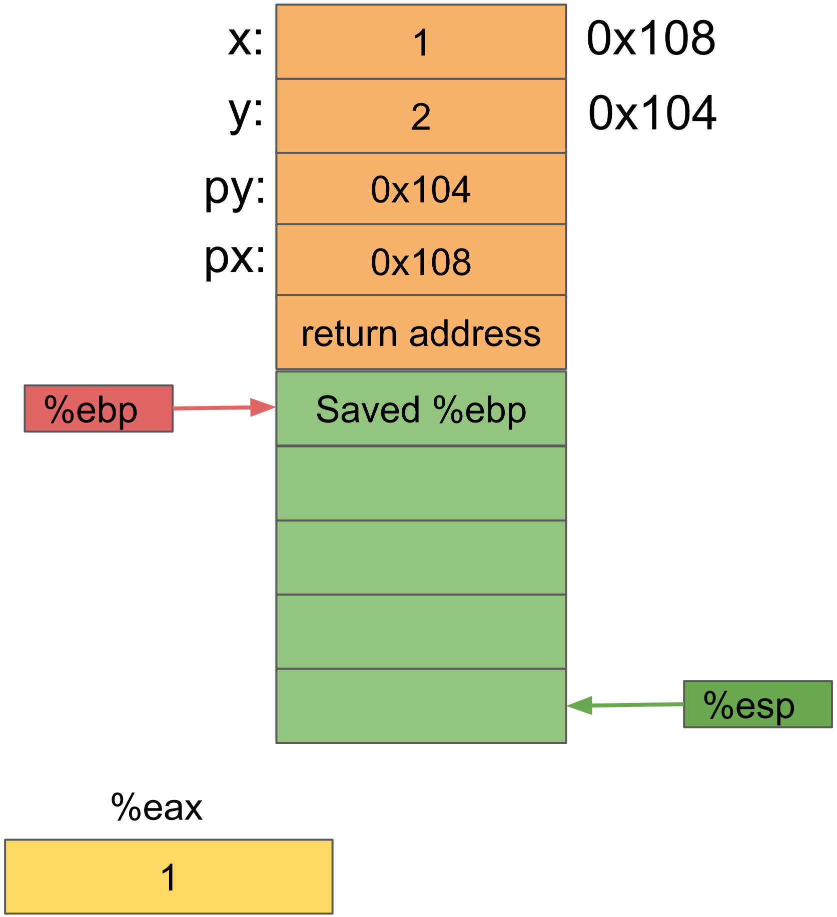
```
swap:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $16, %esp
    movl     8(%ebp), %eax
    movl     (%eax), %eax
    movl     %eax, -4(%ebp)
    movl     12(%ebp), %eax
    movl     (%eax), %eax
    movl     %eax, -8(%ebp)
```

(%eax) = M[0x108] = 1

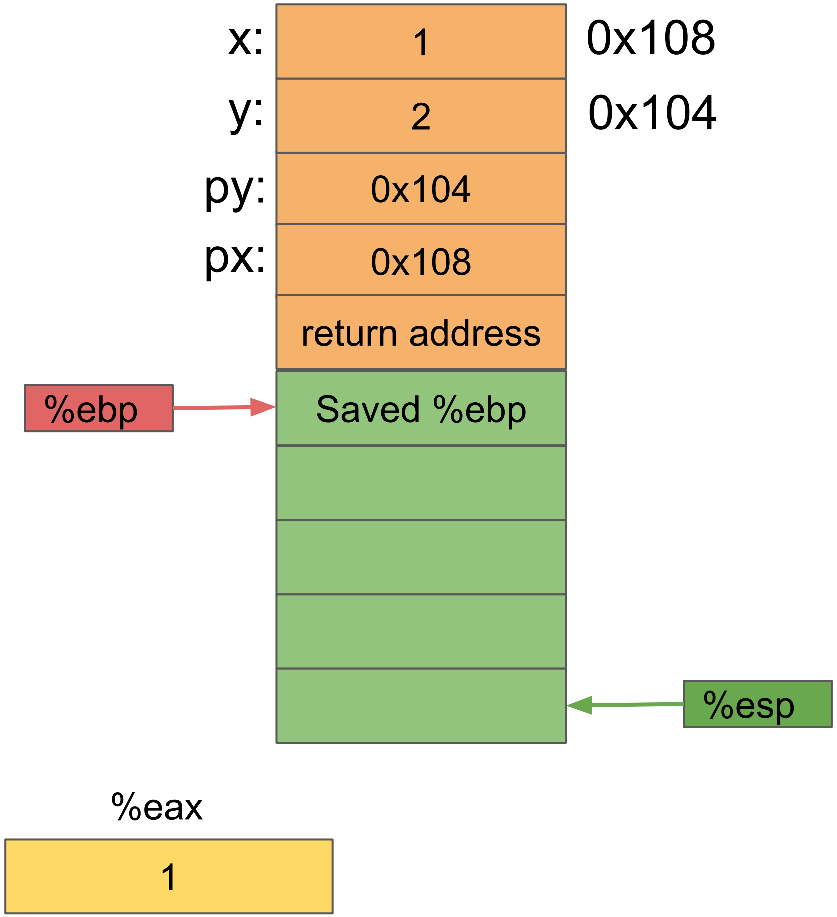
```
    movl     12(%ebp), %eax
    movl     -4(%ebp), %edx
    movl     %edx, (%eax)
    leave
    ret
```



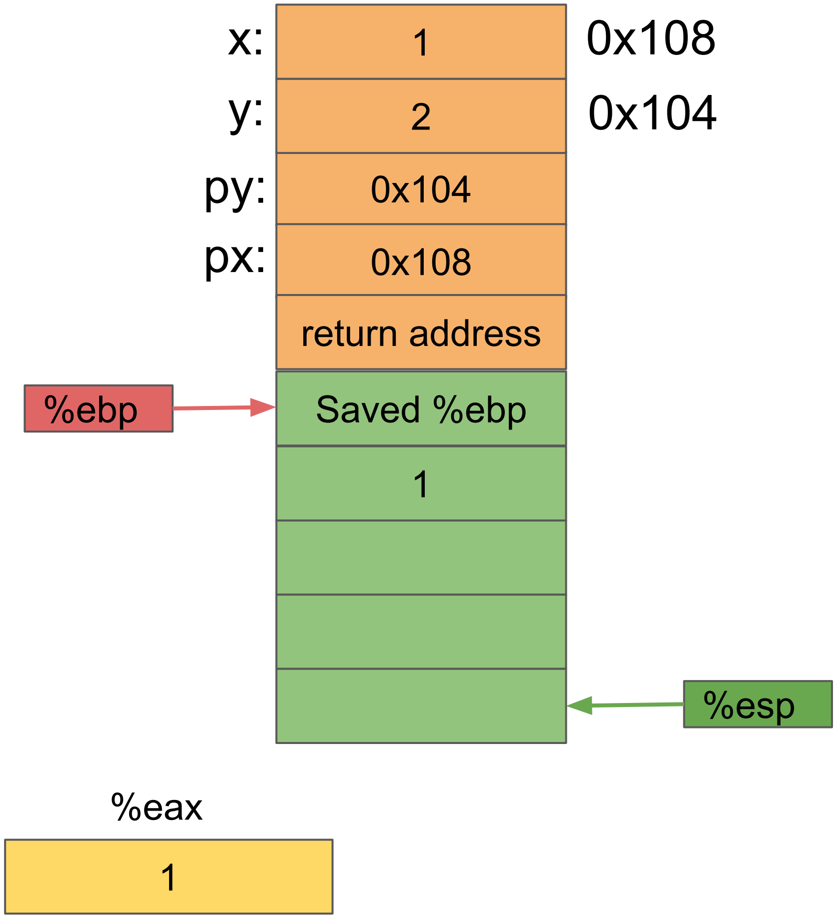
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



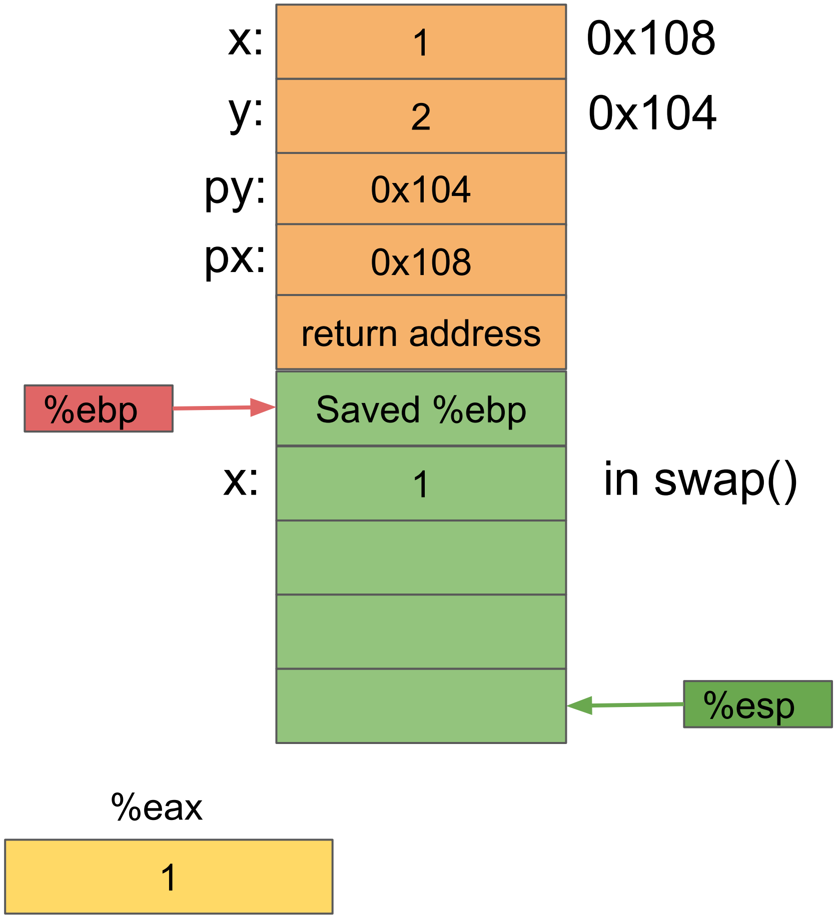
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



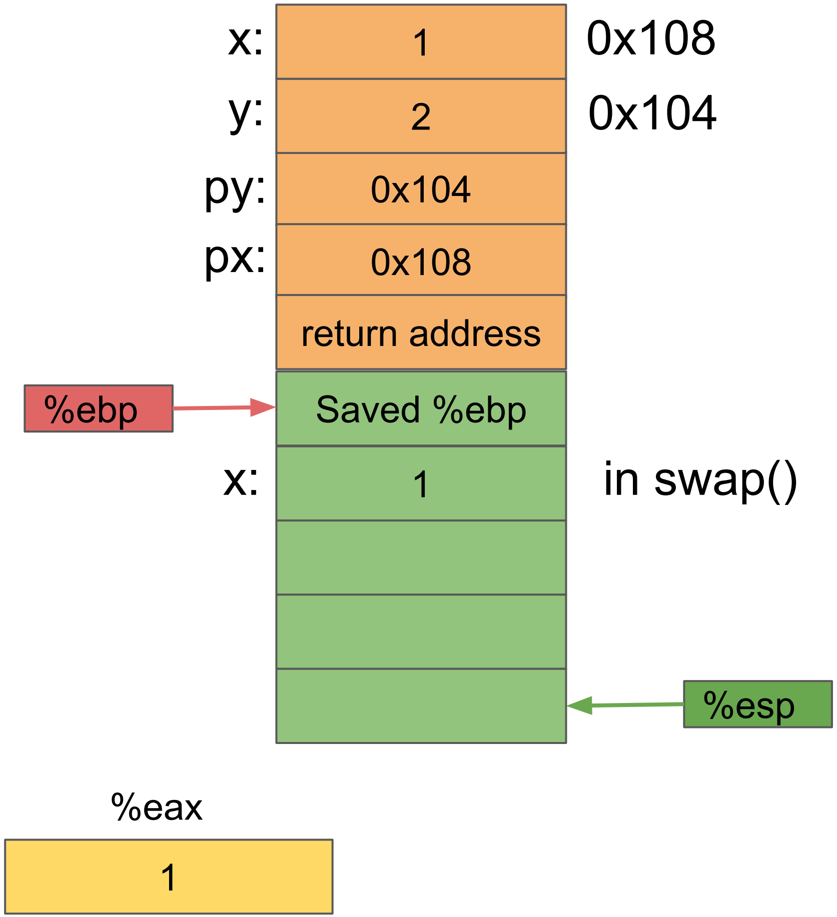
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



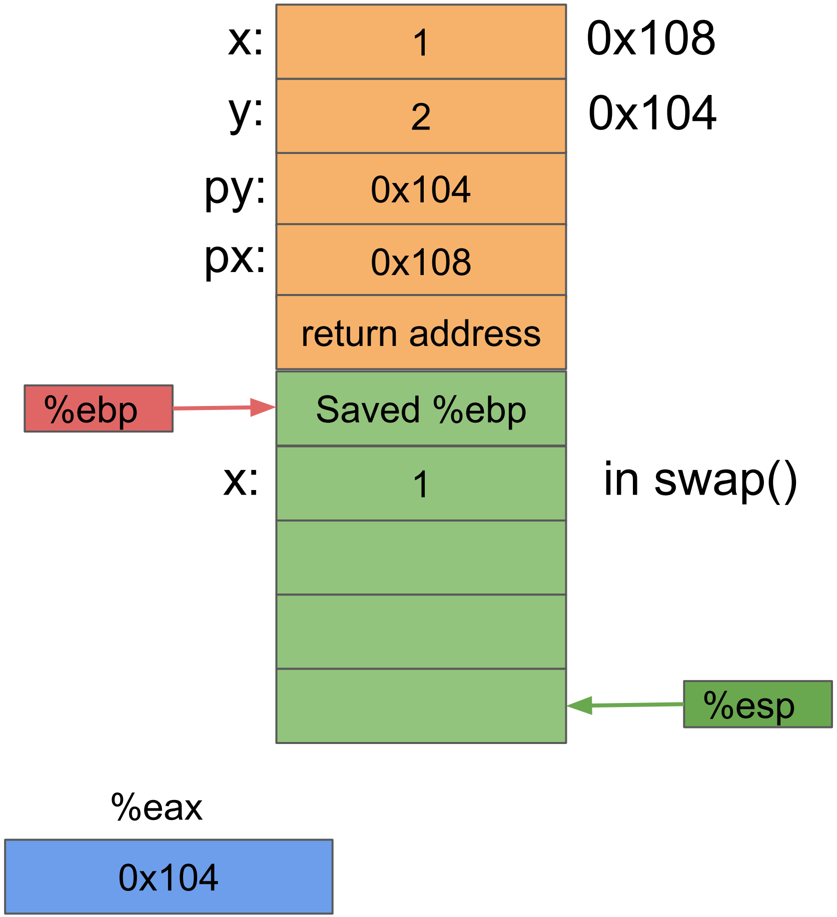
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



x: 1 0x108

y: 2 0x104

py: 0x104

px: 0x108

return address

%ebp

Saved %ebp

x: 1

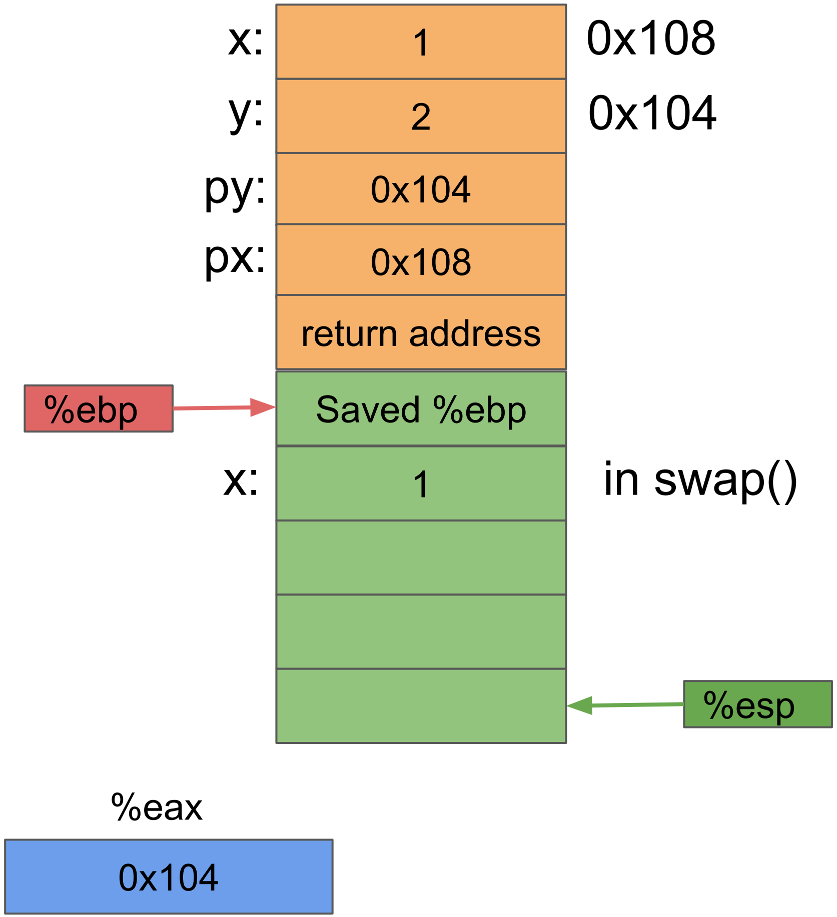
in swap()

%esp

%eax

0x104

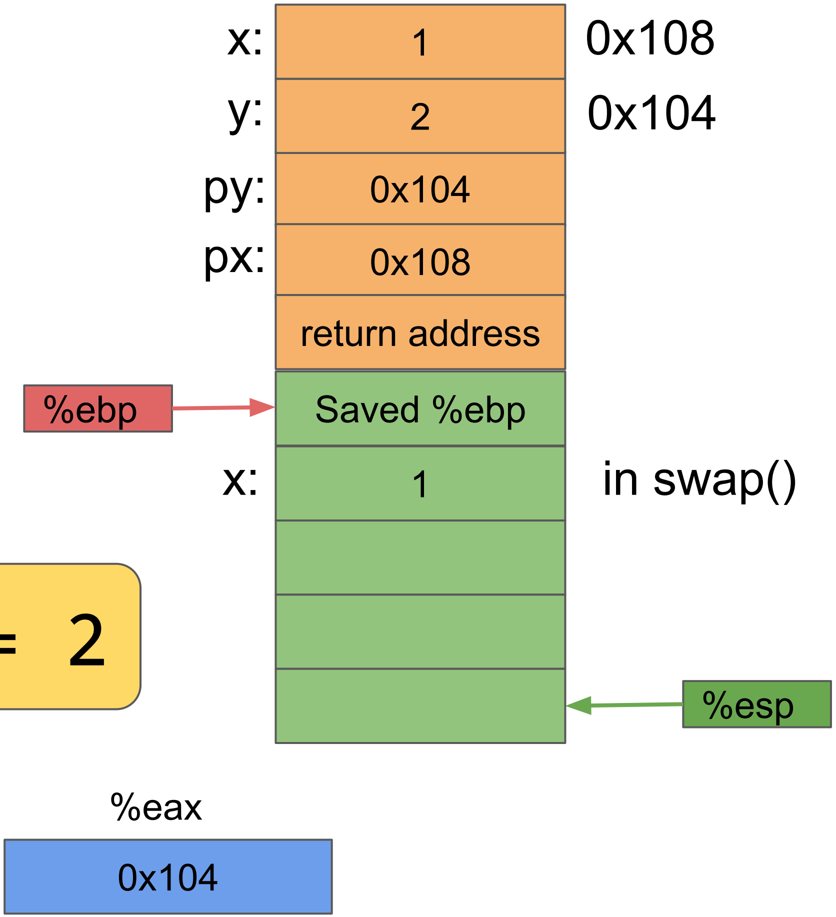
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
```

(%eax) = M[0x104] = 2

```
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```

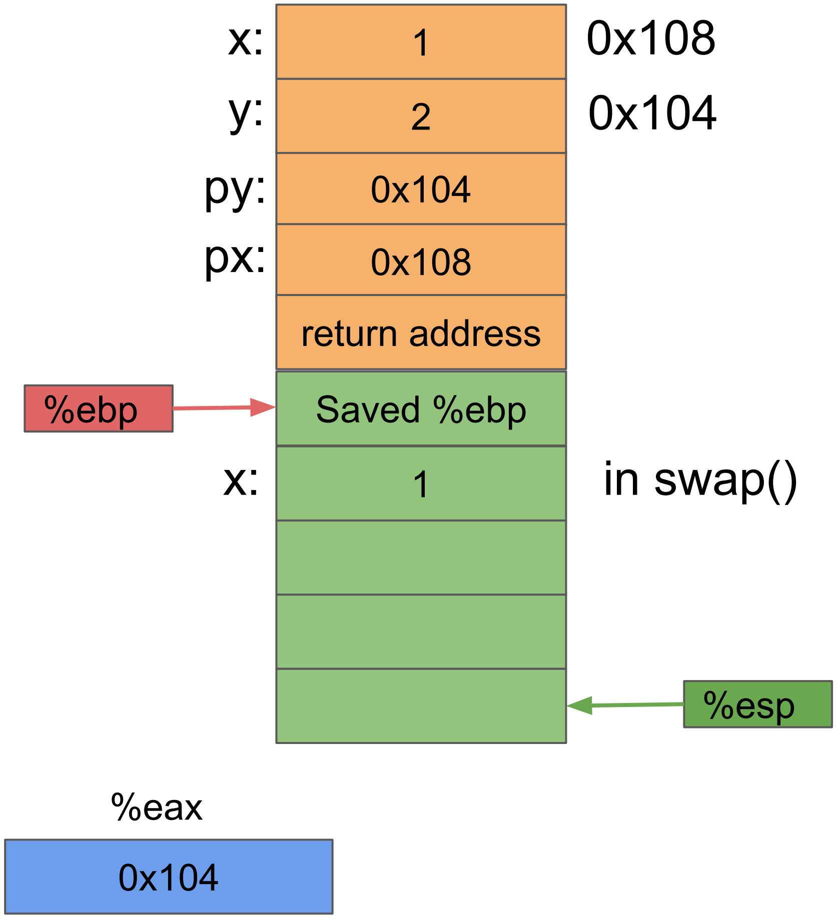


x:	1	0x108
y:	2	0x104
py:	0x104	
px:	0x108	
	return address	
%ebp →	Saved %ebp	
x:	1	in swap()
		← %esp

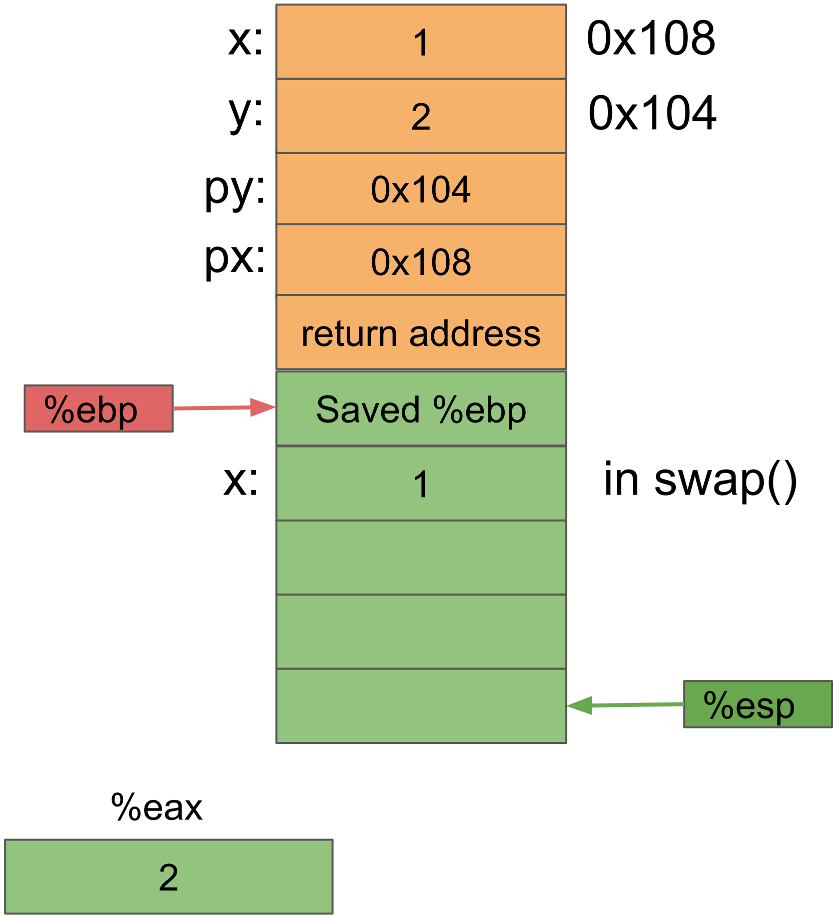
%eax

0x104

```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



x: 1 0x108

y: 2 0x104

py: 0x104

px: 0x108

return address

%ebp

Saved %ebp

x: 1

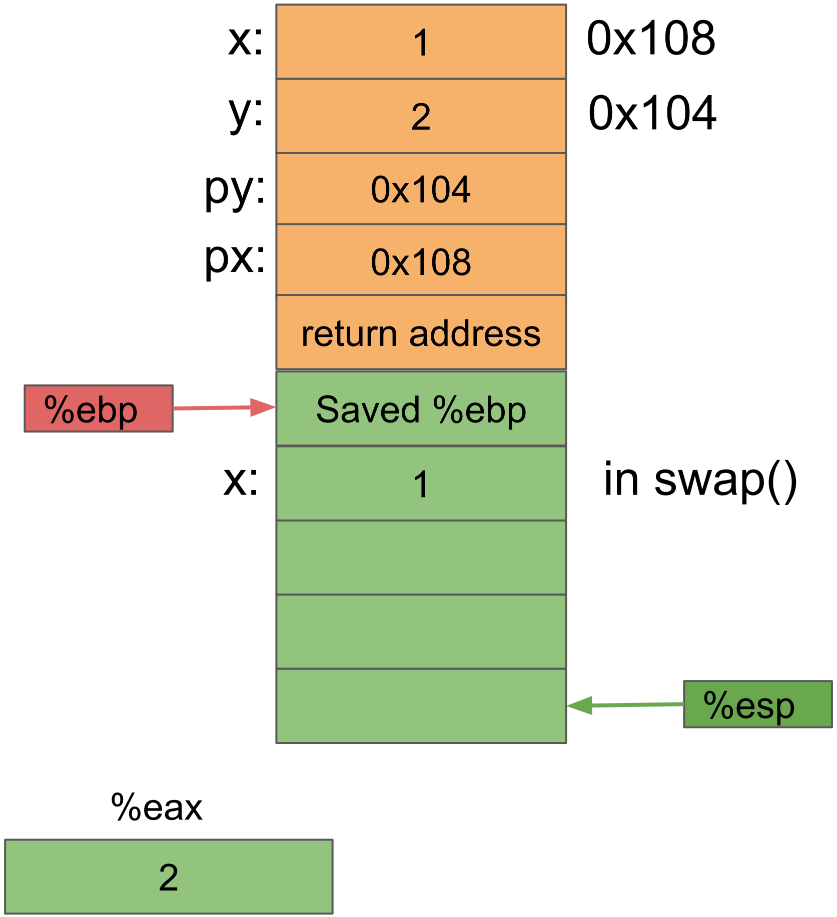
in swap()

%esp

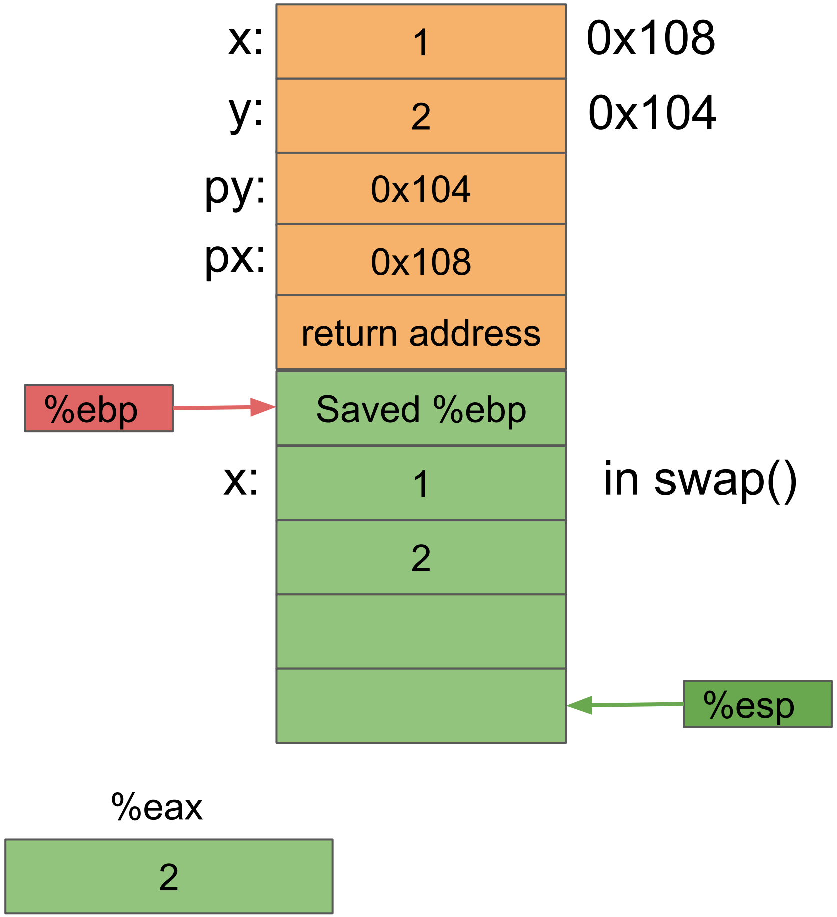
%eax

2

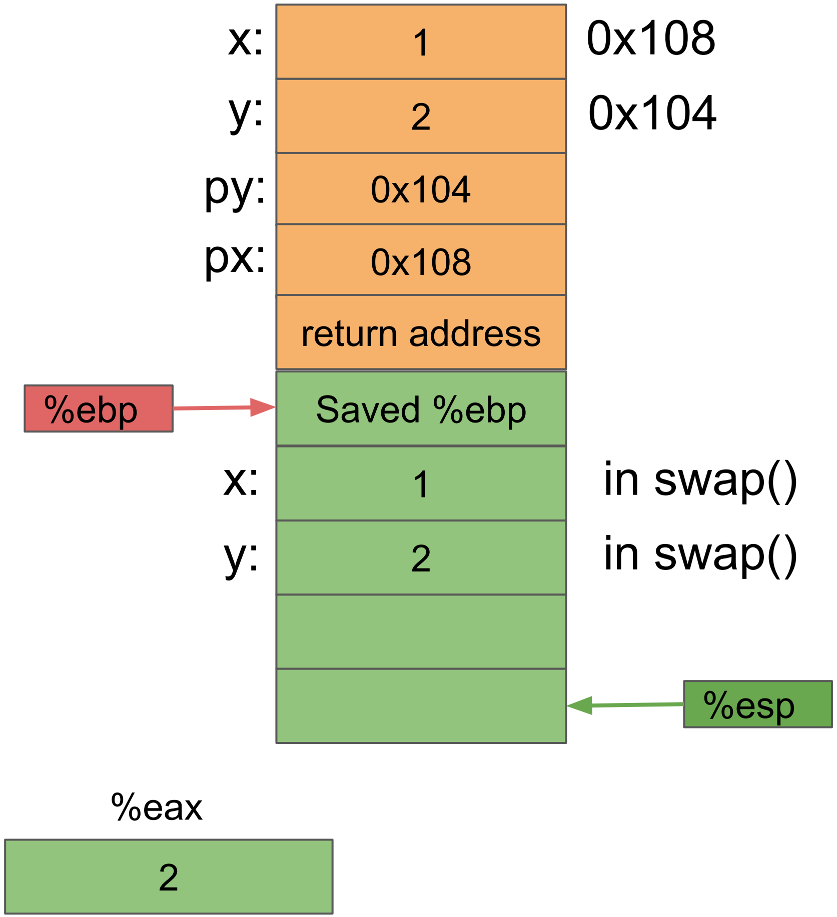
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



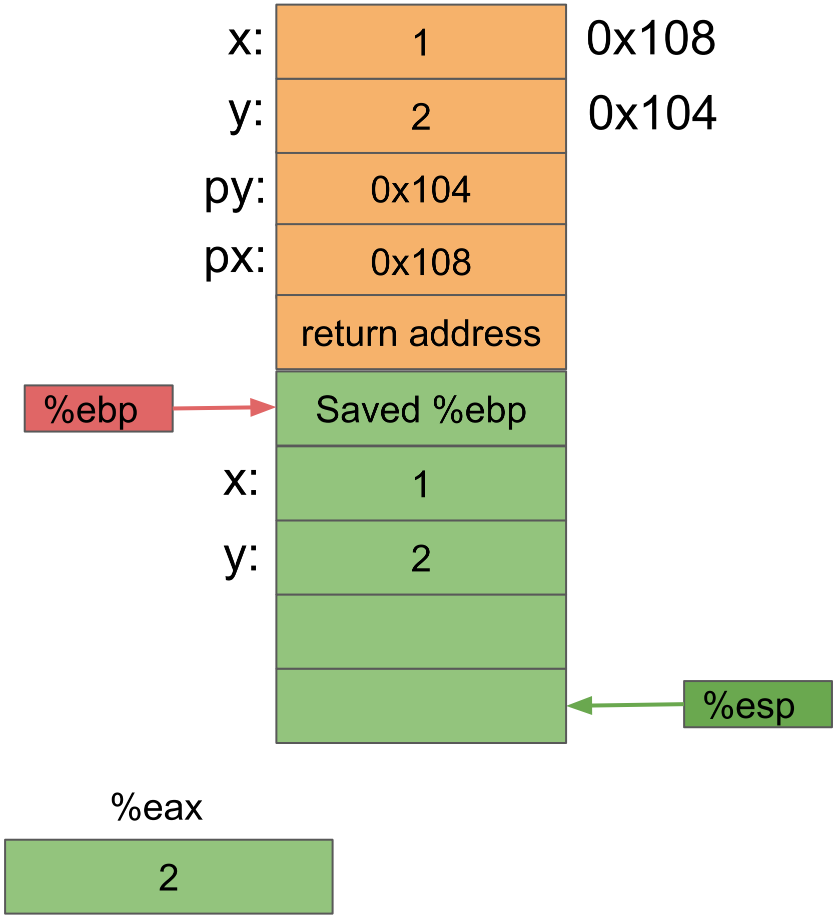
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



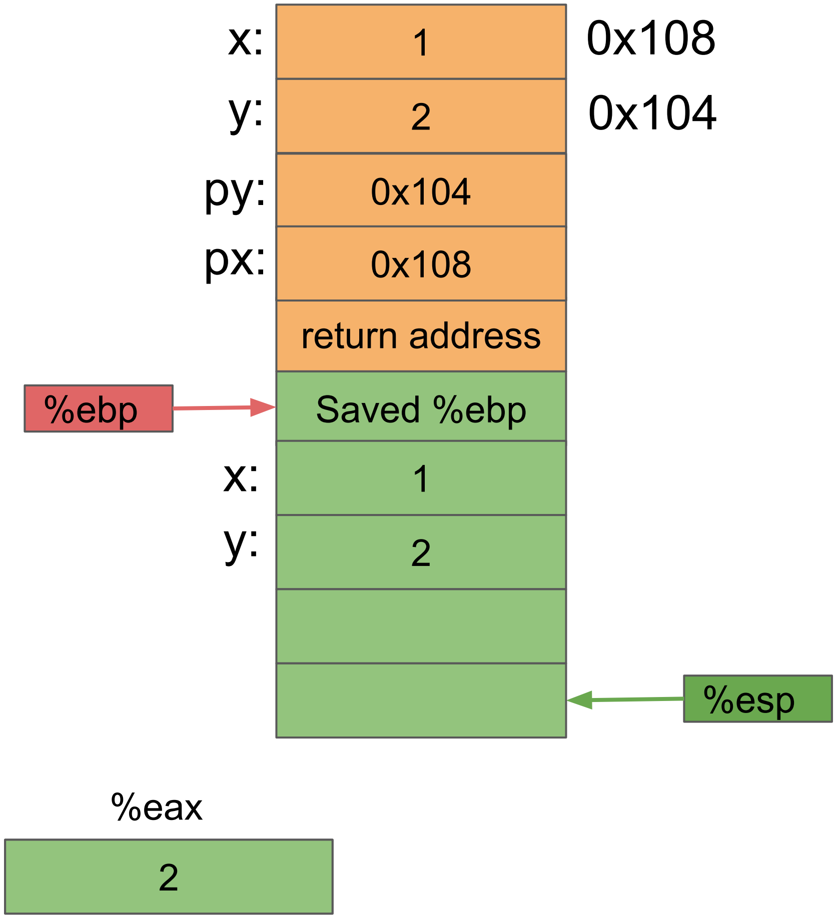
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



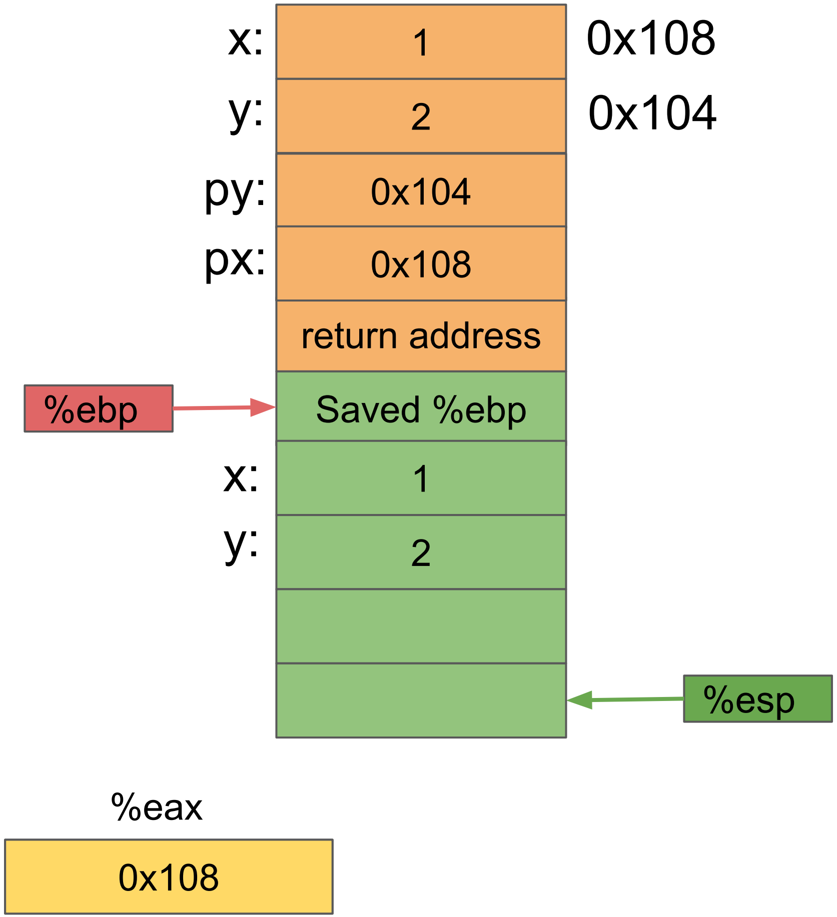
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $16, %esp
    movl     8(%ebp), %eax
    movl     (%eax), %eax
    movl     %eax, -4(%ebp)
    movl     12(%ebp), %eax
    movl     (%eax), %eax
    movl     %eax, -8(%ebp)
    movl     8(%ebp), %eax
    movl     -8(%ebp), %edx
    movl     %edx, (%eax)
    movl     12(%ebp), %eax
    movl     -4(%ebp), %edx
    movl     %edx, (%eax)
    leave
    ret
```



x: 1 0x108

y: 2 0x104

py: 0x104

px: 0x108

return address

%ebp Saved %ebp

x: 1

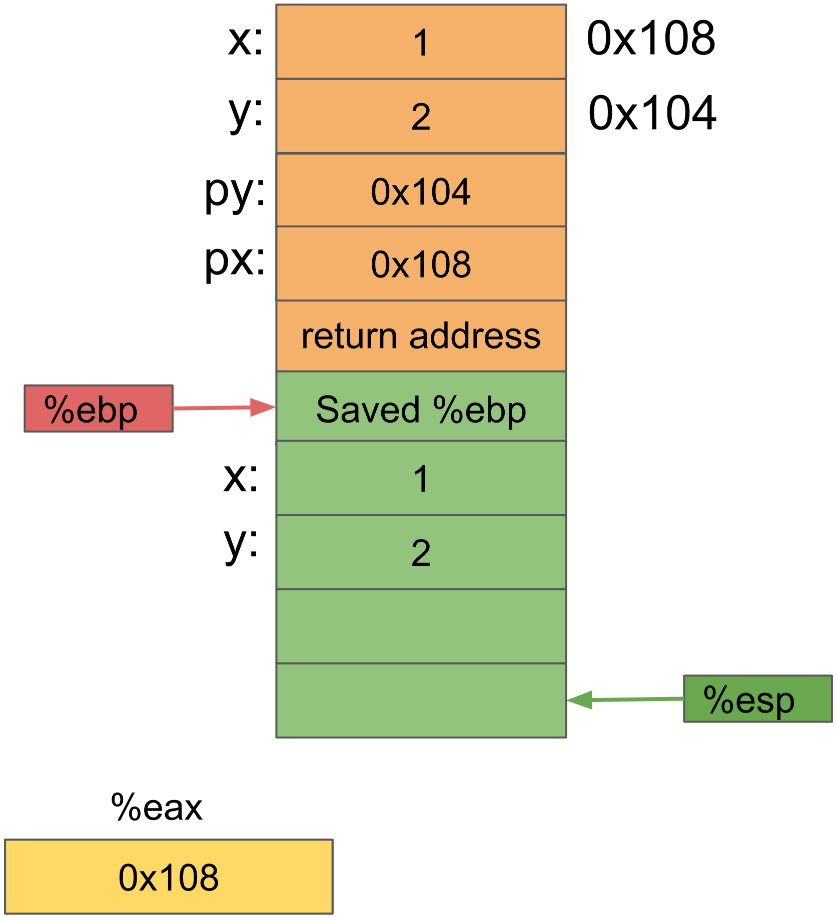
y: 2

%esp

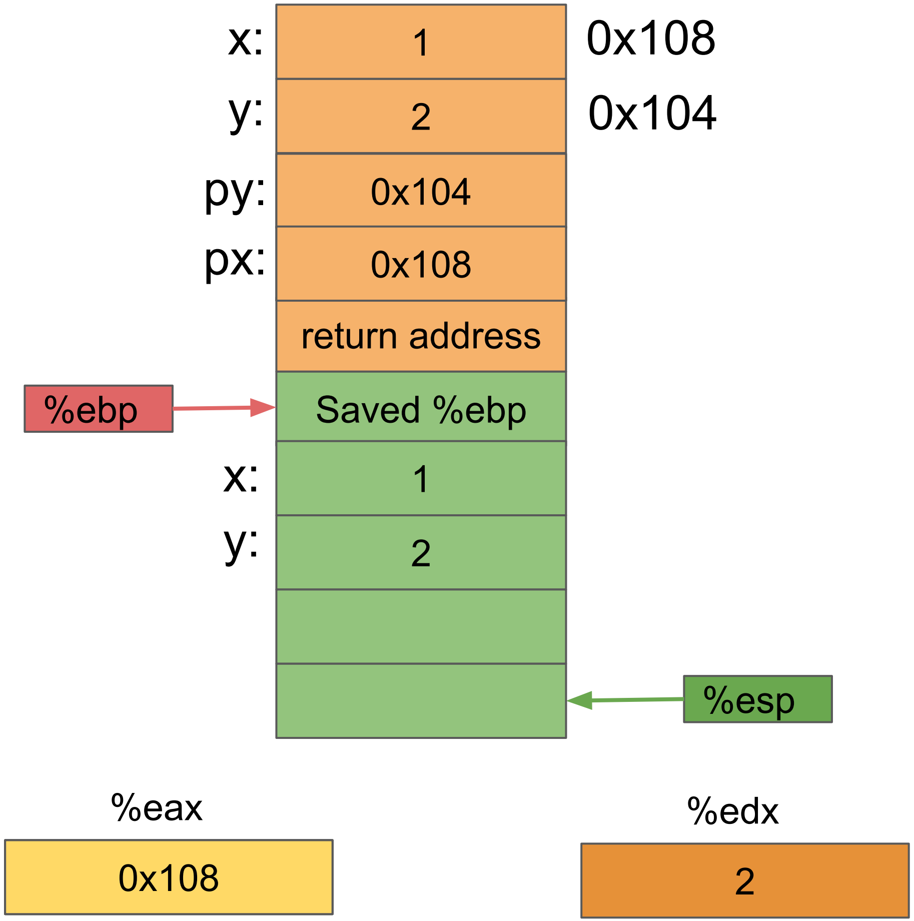
%eax

0x108

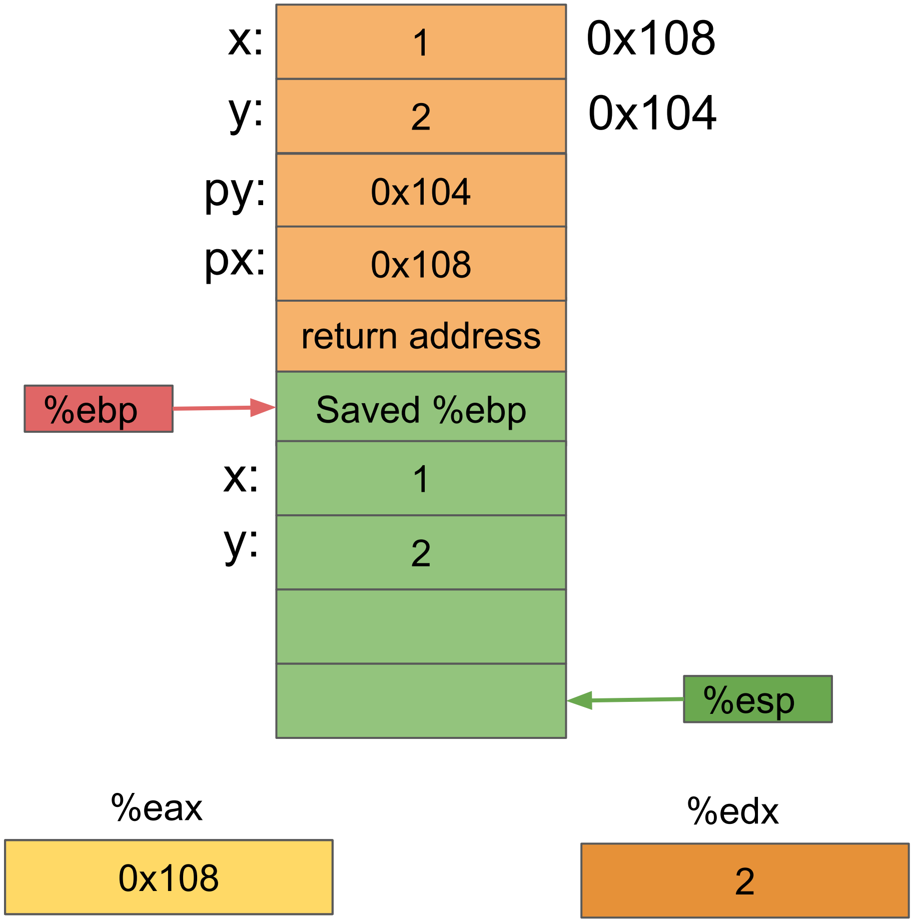
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```

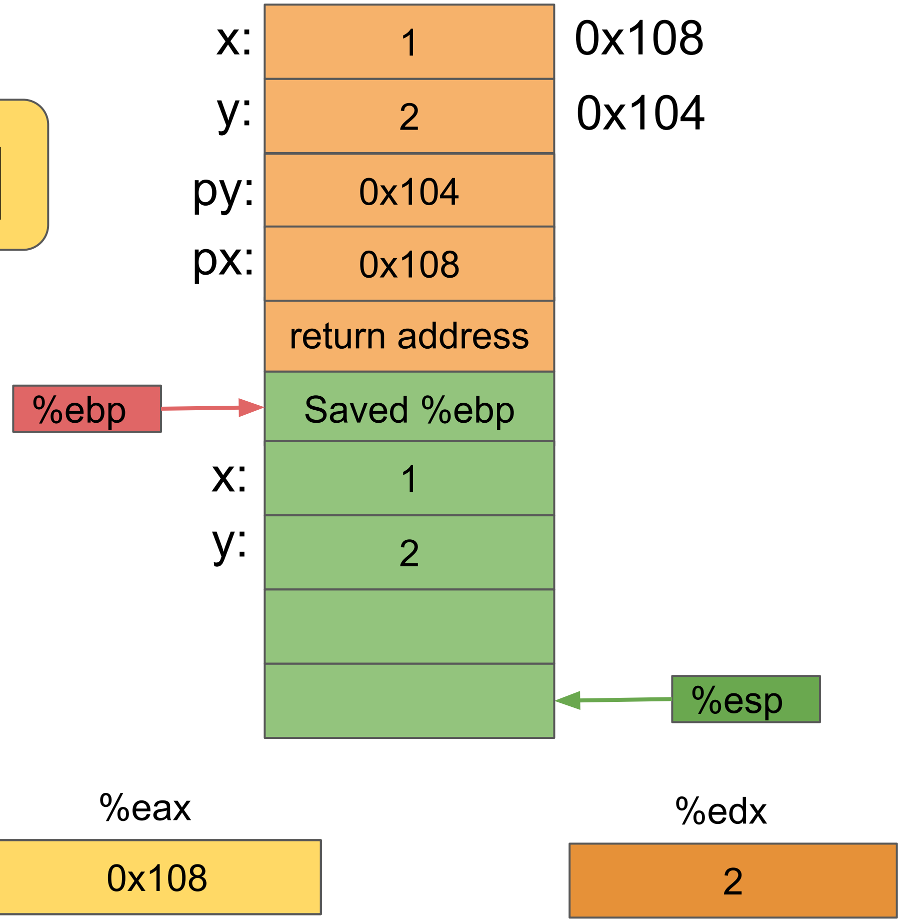


```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```

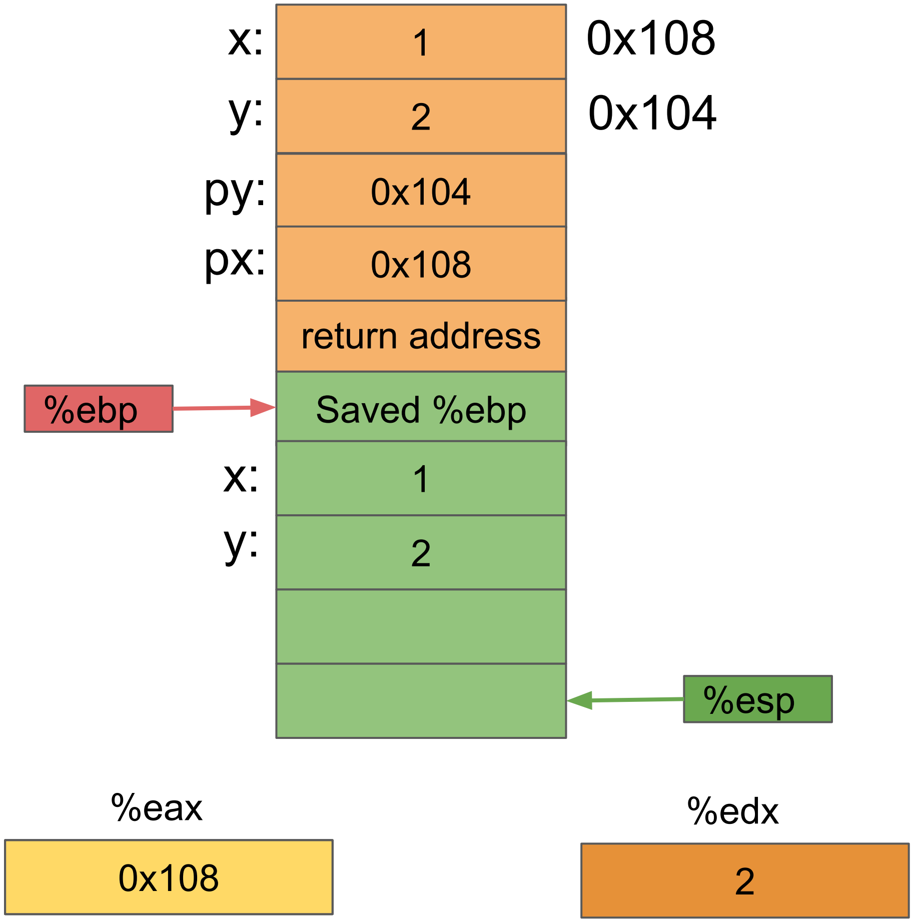


```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    su
    mo
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```

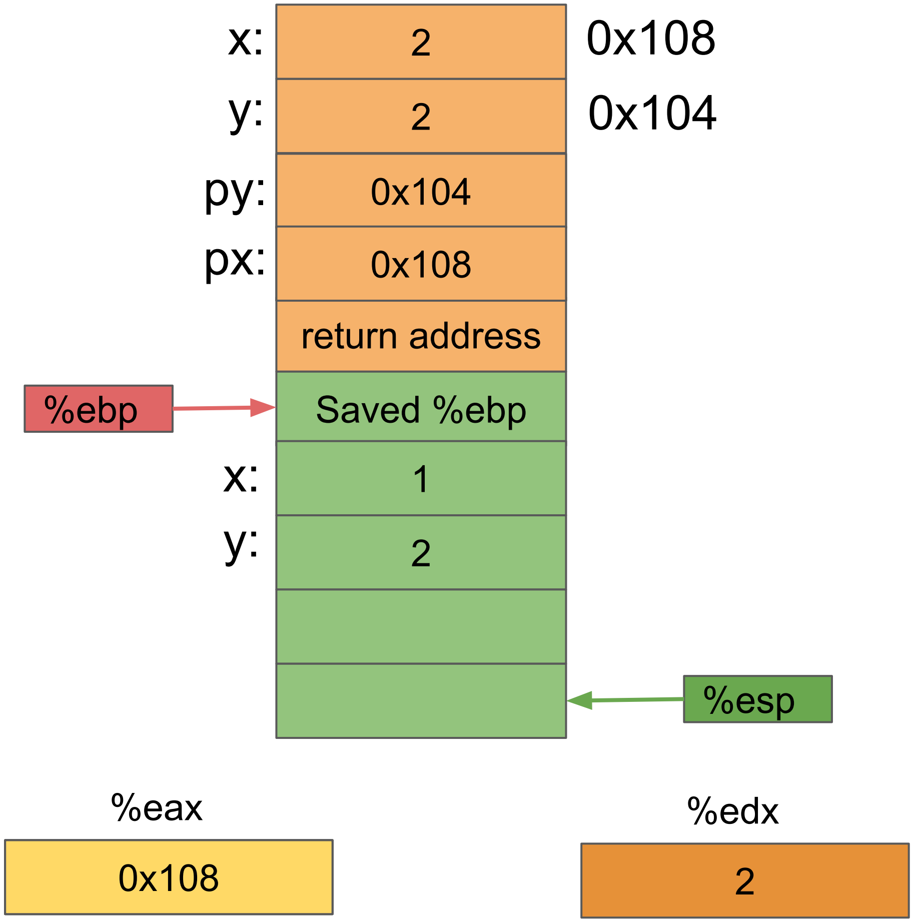
(%eax) = M[0x108]



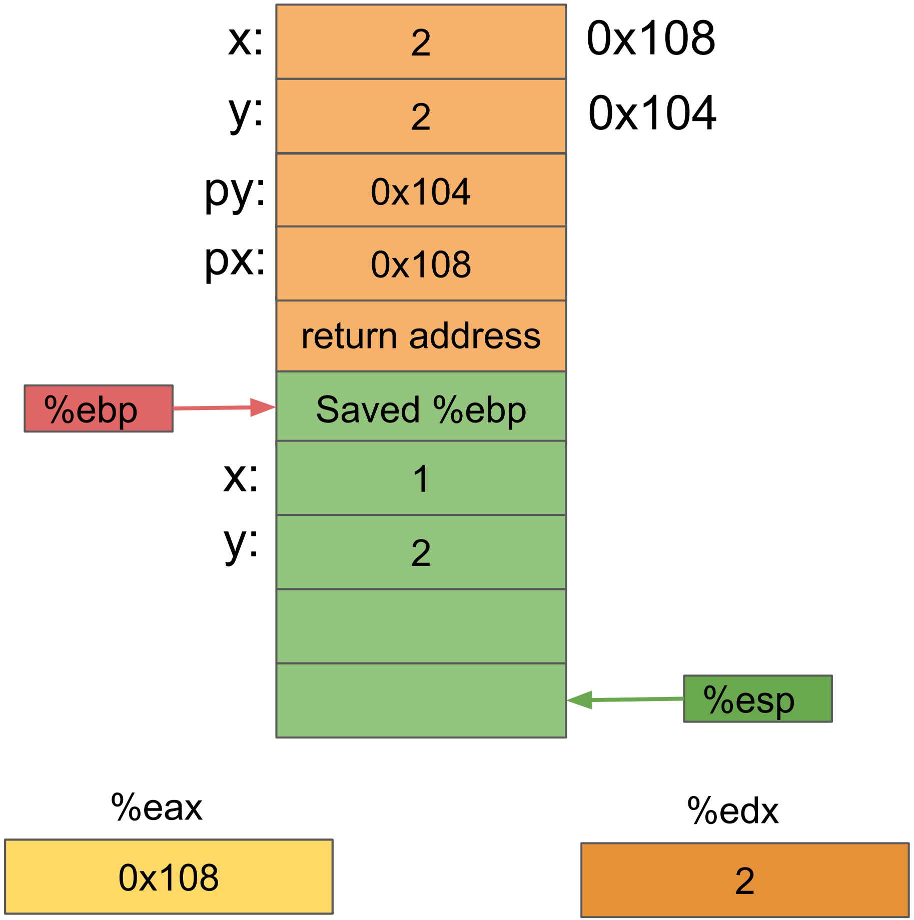
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



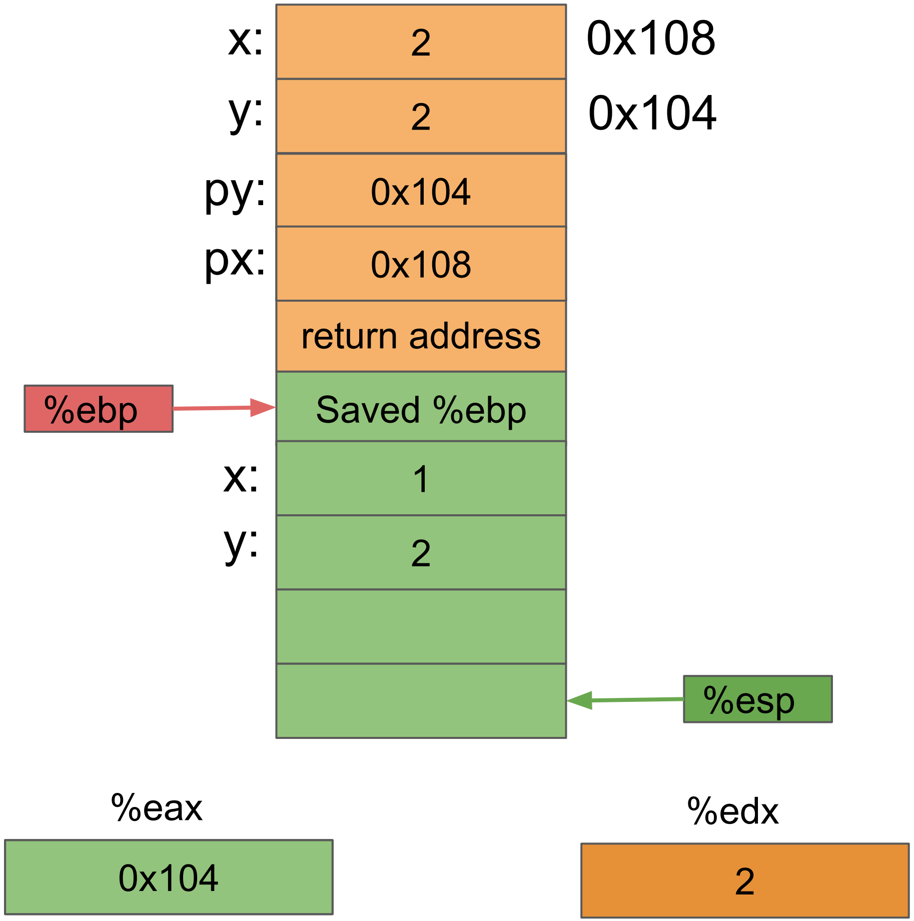
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



x: 2 0x108

y: 2 0x104

py: 0x104

px: 0x108

return address

%ebp

Saved %ebp

x: 1

y: 2

%esp

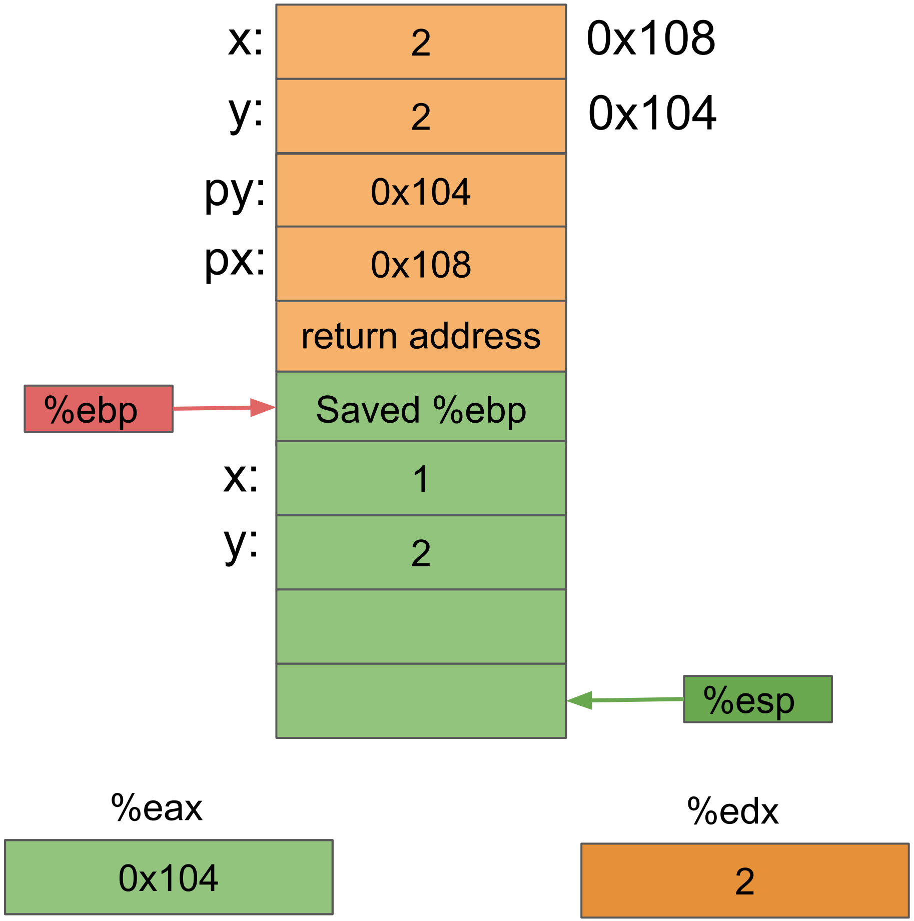
%eax

0x104

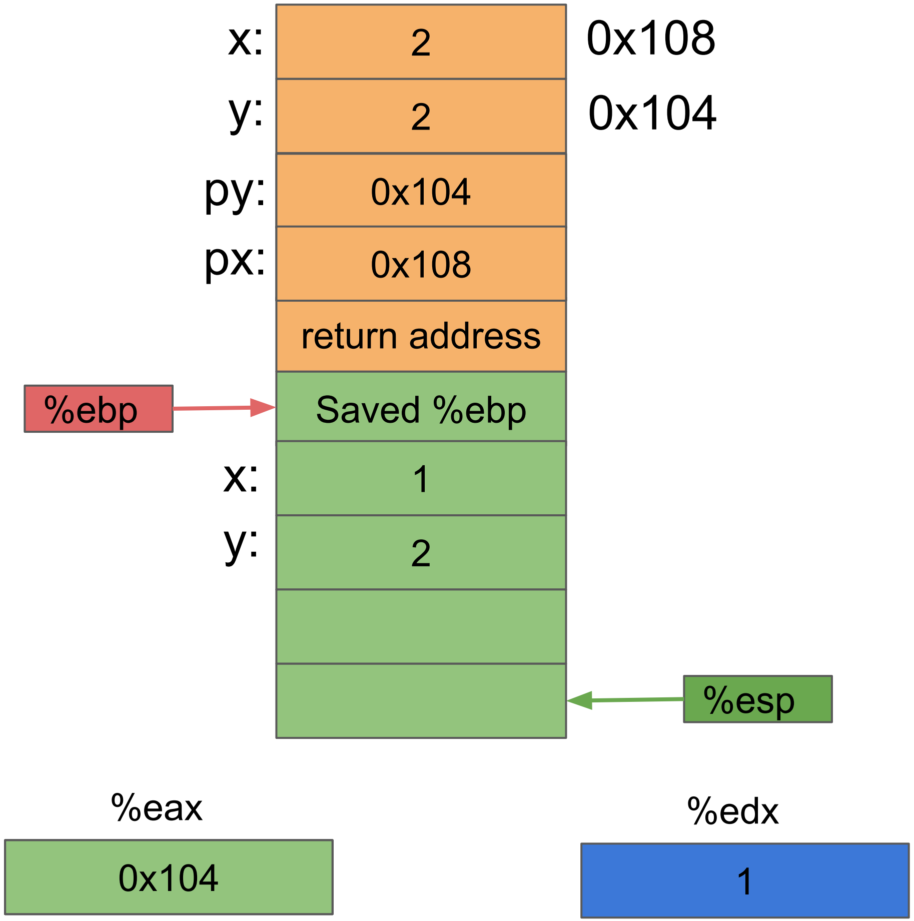
%edx

2

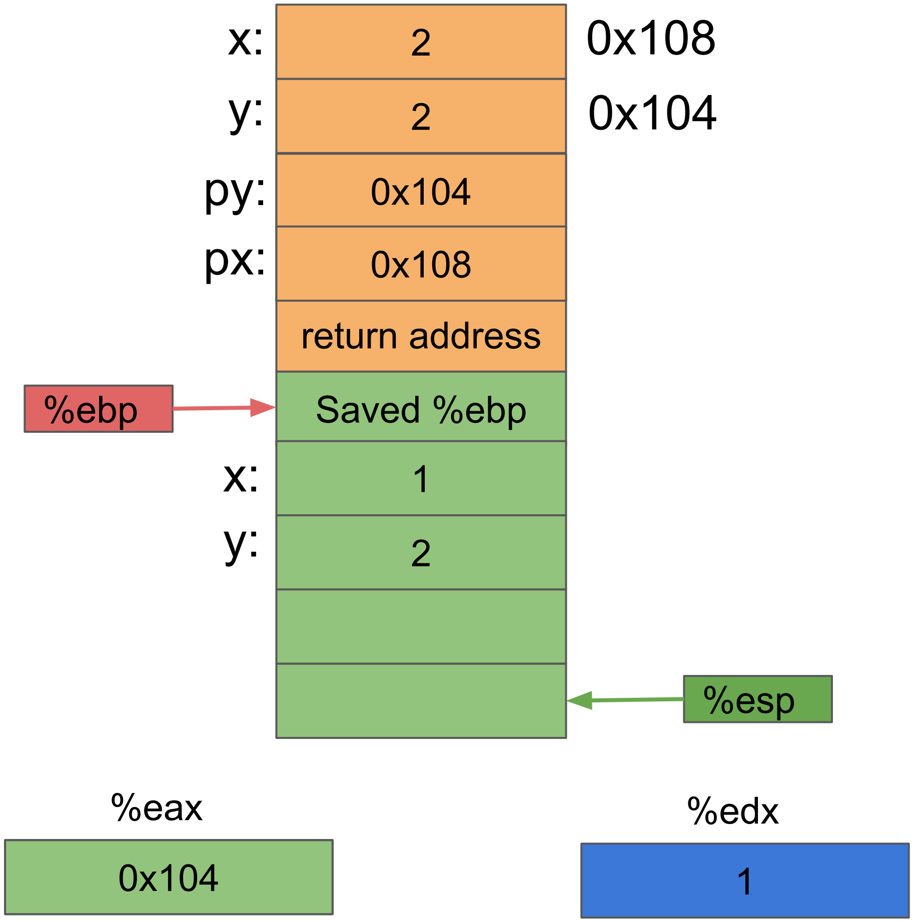
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $16, %esp
    movl     8(%ebp), %eax
    movl     (%eax), %eax
    movl     %eax, -4(%ebp)
    movl     12(%ebp), %eax
    movl     (%eax), %eax
    movl     %eax, -8(%ebp)
    movl     8(%ebp), %eax
    movl     -8(%ebp), %edx
    movl     %edx, (%eax)
    movl     12(%ebp), %eax
    movl     -4(%ebp), %edx
    movl     %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



x: 2 0x108

y: 2 0x104

py: 0x104

px: 0x108

return address

%ebp → Saved %ebp

x: 1

y: 2

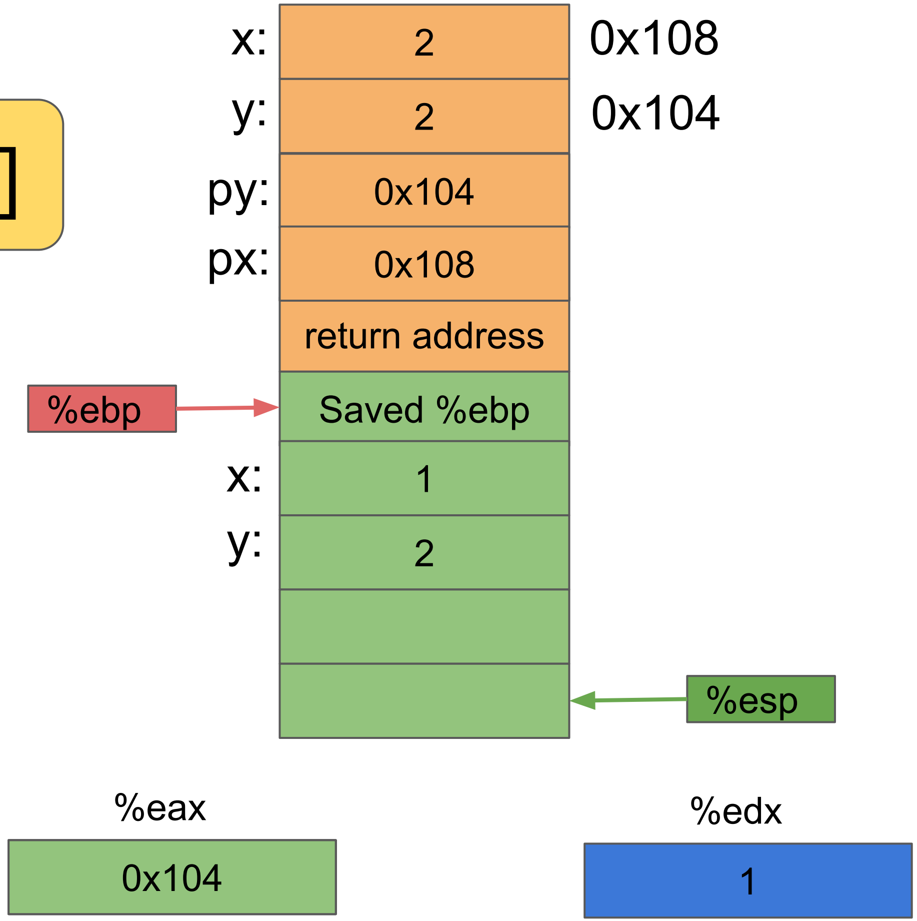
%esp

%eax 0x104

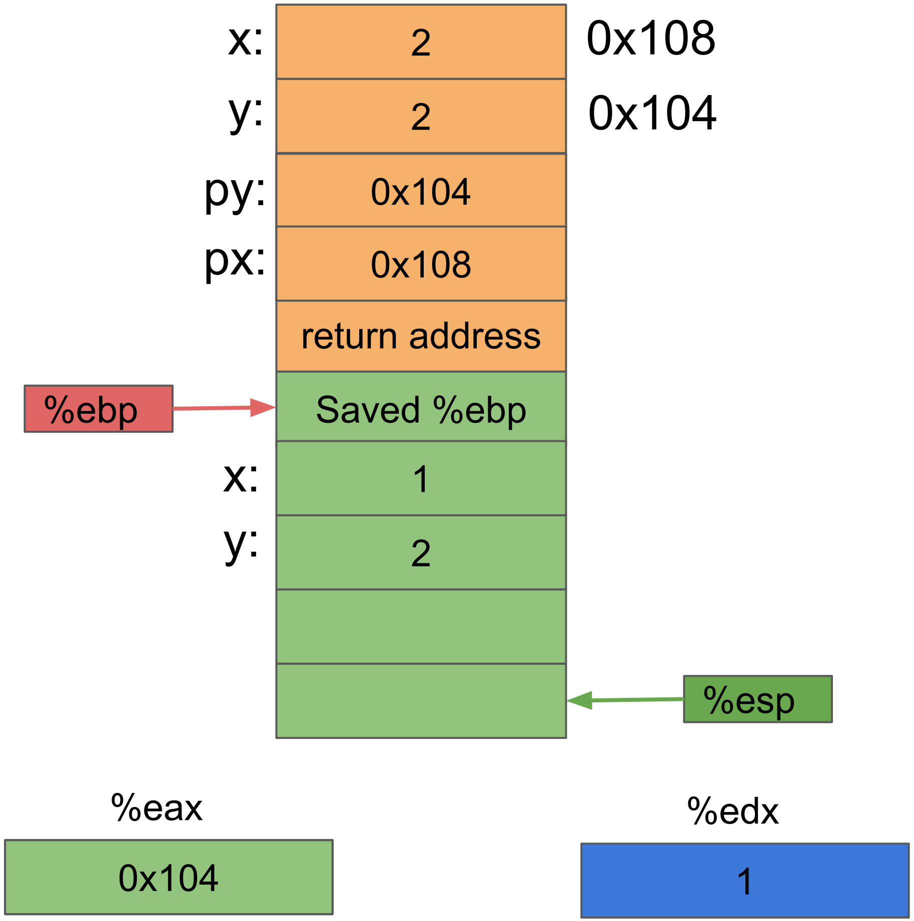
%edx 1

```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    su
    mo
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```

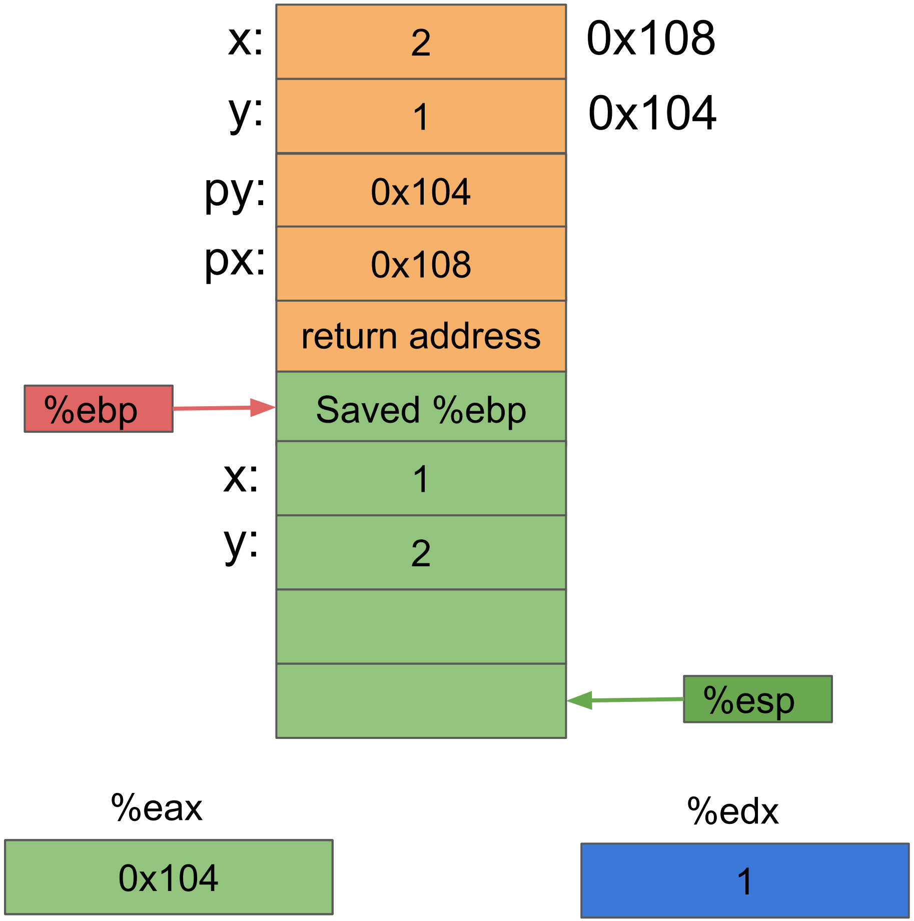
(%eax) = M[0x104]



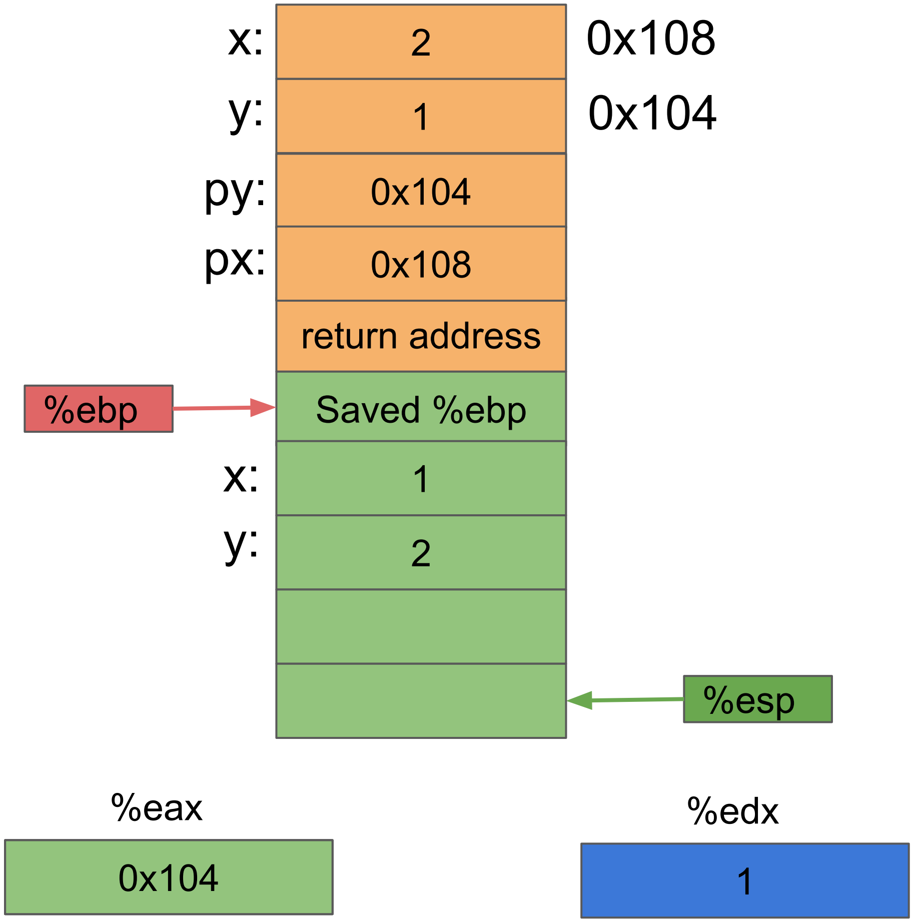
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



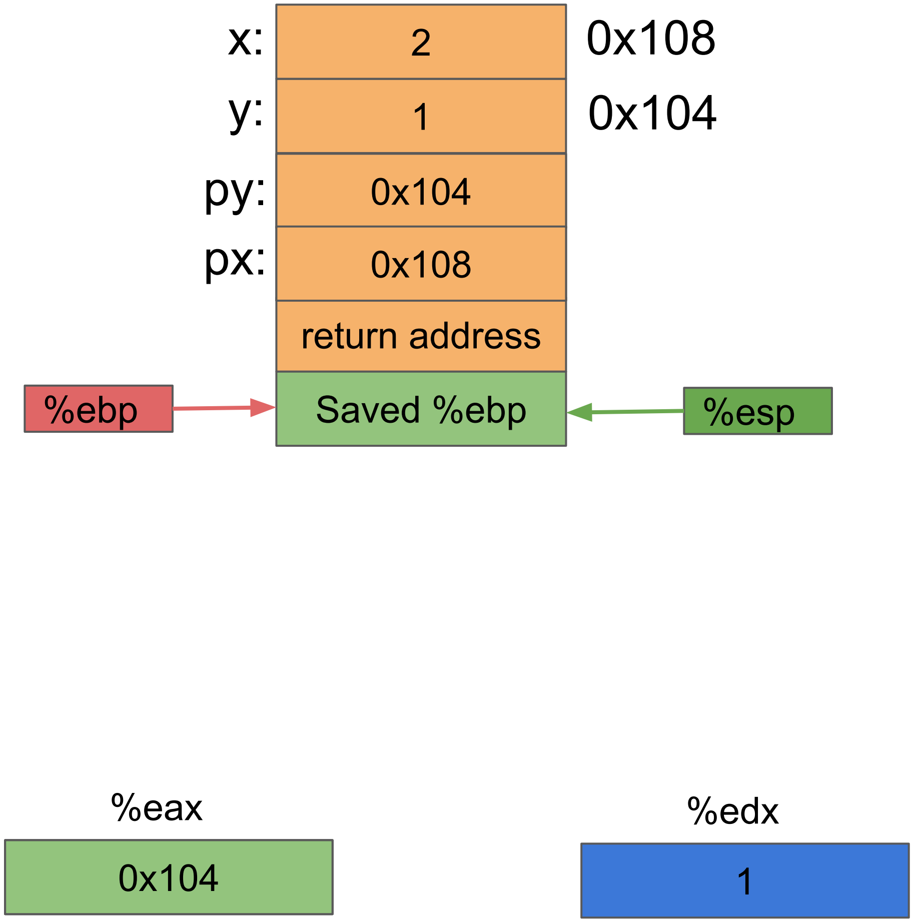
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



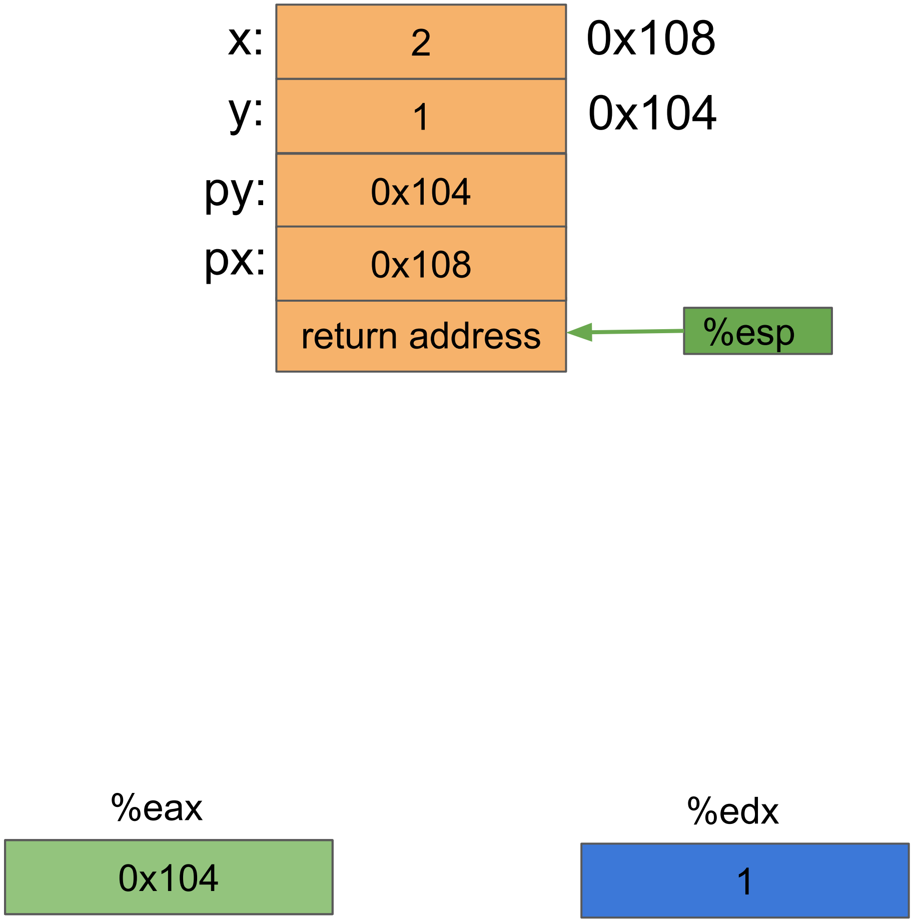
```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



x: 2 0x108

y: 1 0x104

py: 0x104

px: 0x108

return address ← %esp

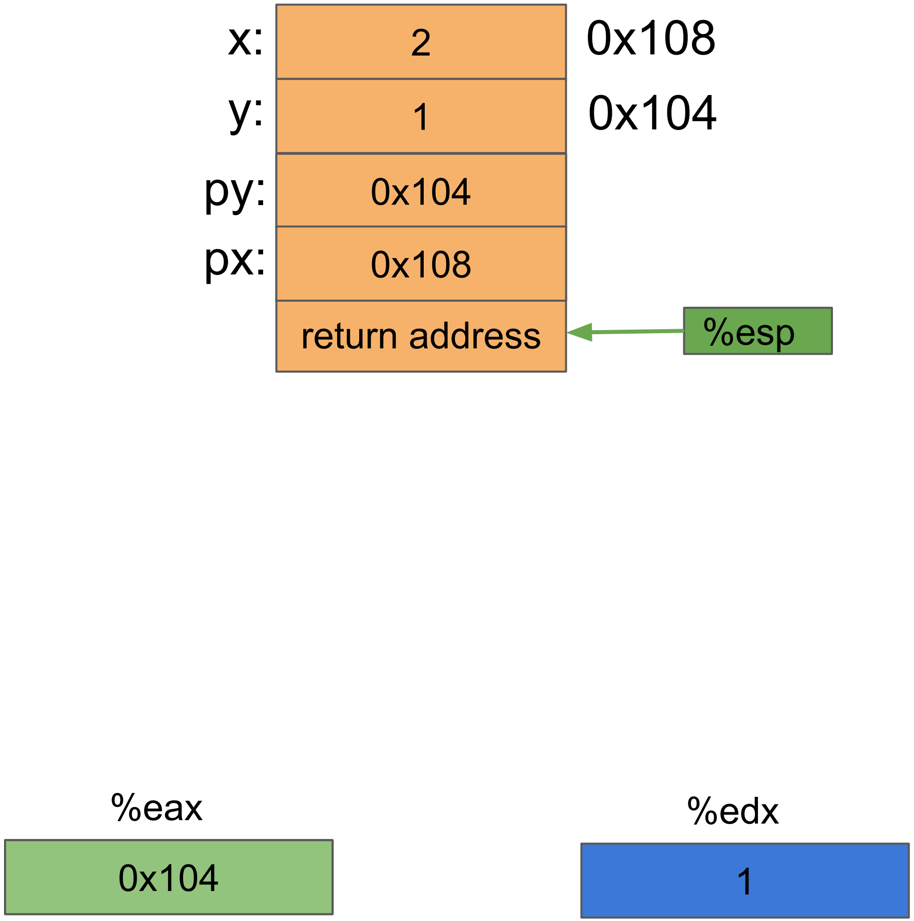
%eax

0x104

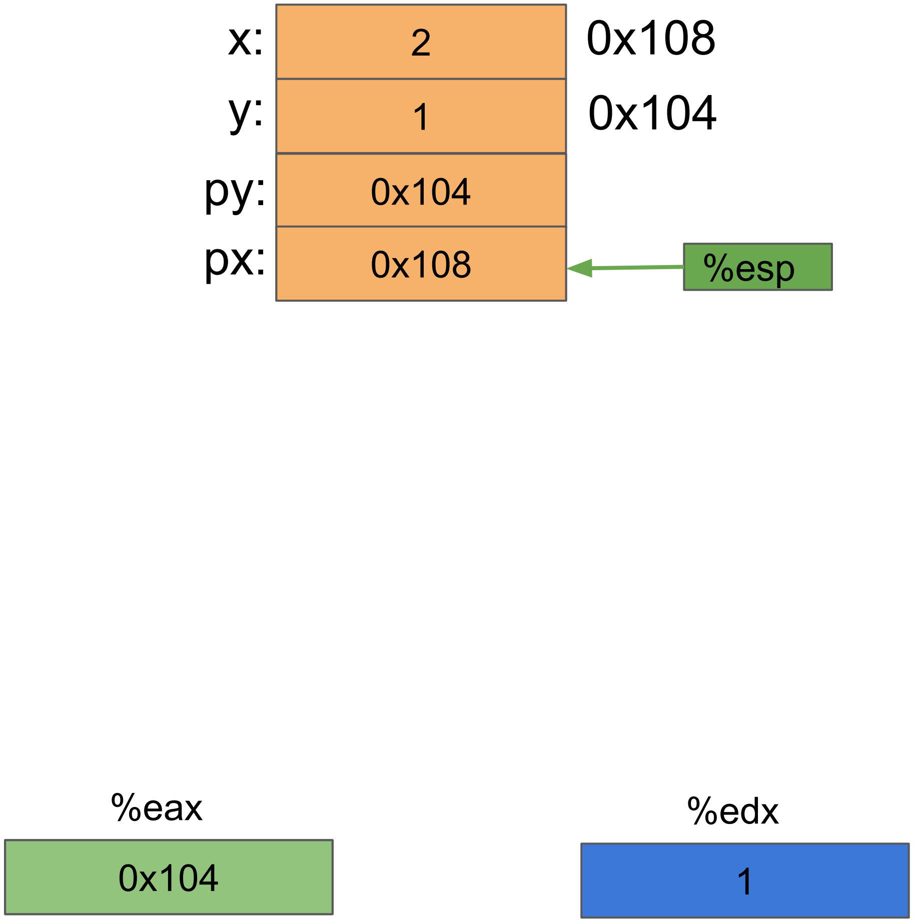
%edx

1

```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



```
swap:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    8(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -4(%ebp)
    movl    12(%ebp), %eax
    movl    (%eax), %eax
    movl    %eax, -8(%ebp)
    movl    8(%ebp), %eax
    movl    -8(%ebp), %edx
    movl    %edx, (%eax)
    movl    12(%ebp), %eax
    movl    -4(%ebp), %edx
    movl    %edx, (%eax)
    leave
    ret
```



x: 2 0x108

y: 1 0x104

py: 0x104

px: 0x108

%esp

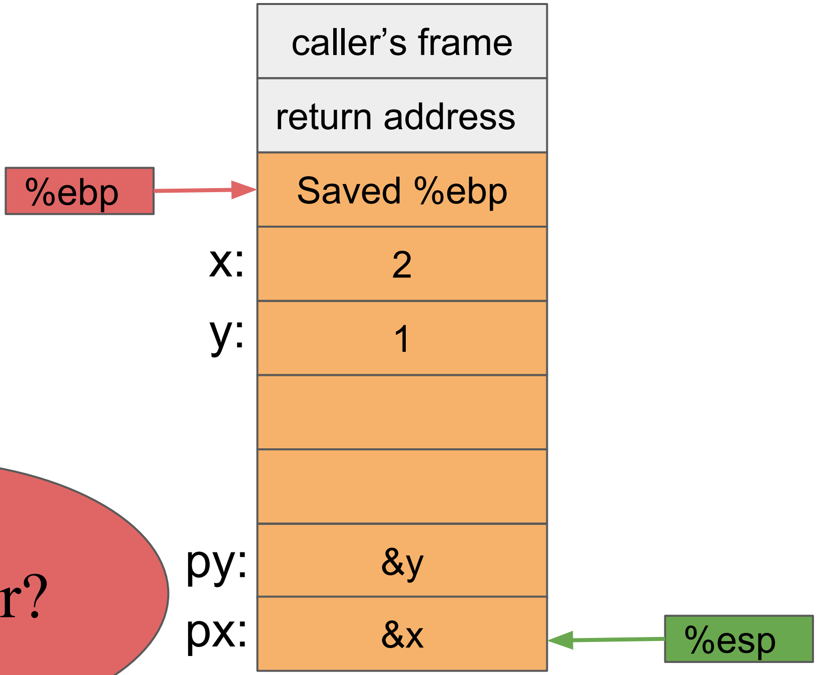
%eax 0x104

%edx 1

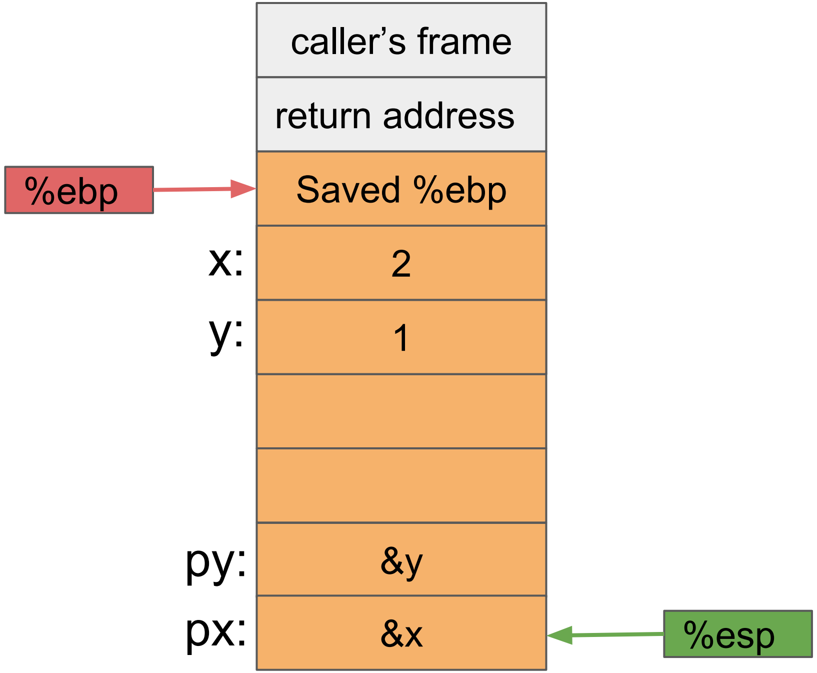
return to main()

```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

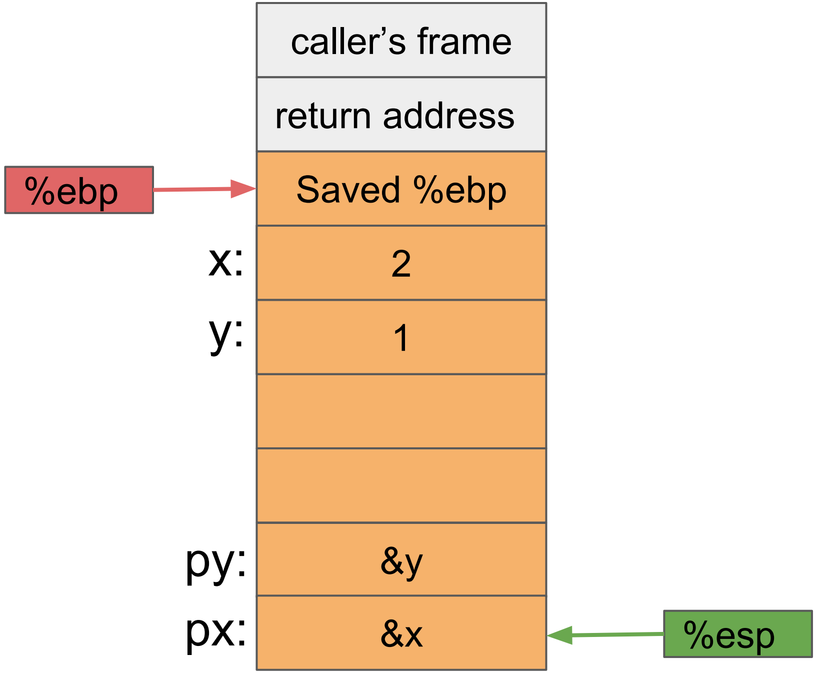
Remember?



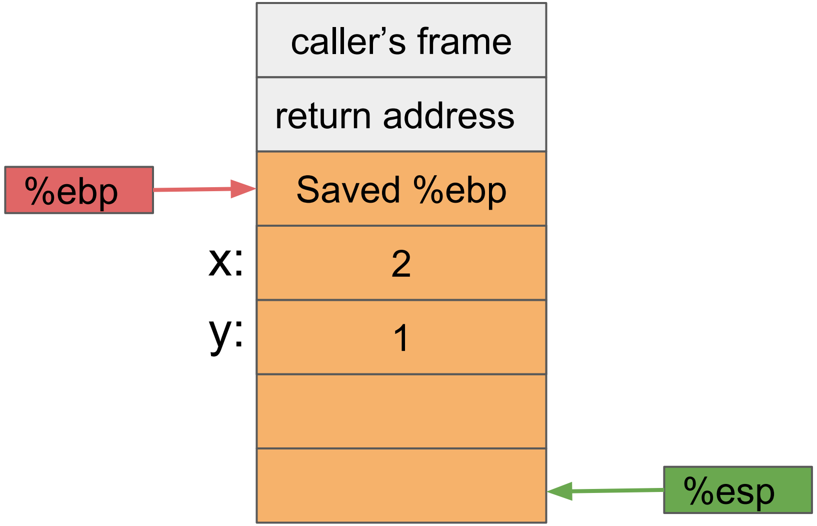
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



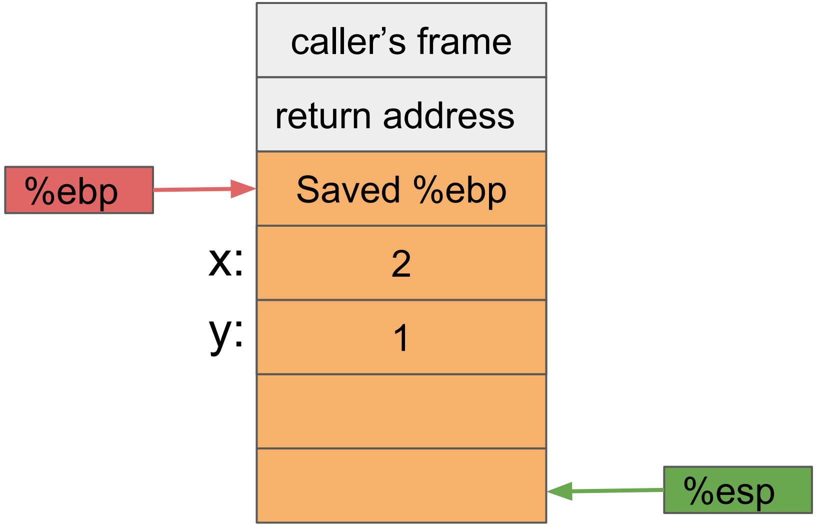
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



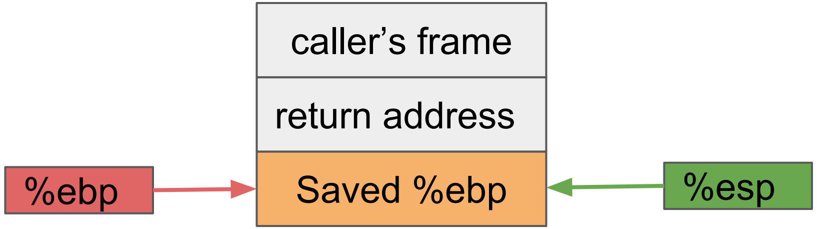
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



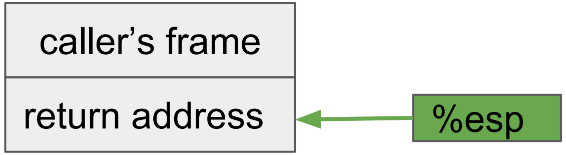
```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



```
main:
     pushl    %ebp
     movl     %esp, %ebp
     subl     $16, %esp
     movl     $1, -4(%ebp)
     movl     $2, -8(%ebp)
     leal     -8(%ebp), %eax
     pushl    %eax
     leal     -4(%ebp), %eax
     pushl    %eax
     call     swap
     addl     $8, %esp
     leave
     ret
```



```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

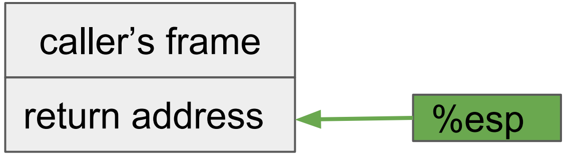


caller’s frame

return address

%esp

```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```

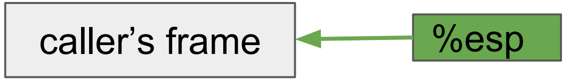


caller’s frame

return address

%esp

```
main:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $16, %esp
    movl    $1, -4(%ebp)
    movl    $2, -8(%ebp)
    leal    -8(%ebp), %eax
    pushl   %eax
    leal    -4(%ebp), %eax
    pushl   %eax
    call    swap
    addl    $8, %esp
    leave
    ret
```



finish

ret

Questions?