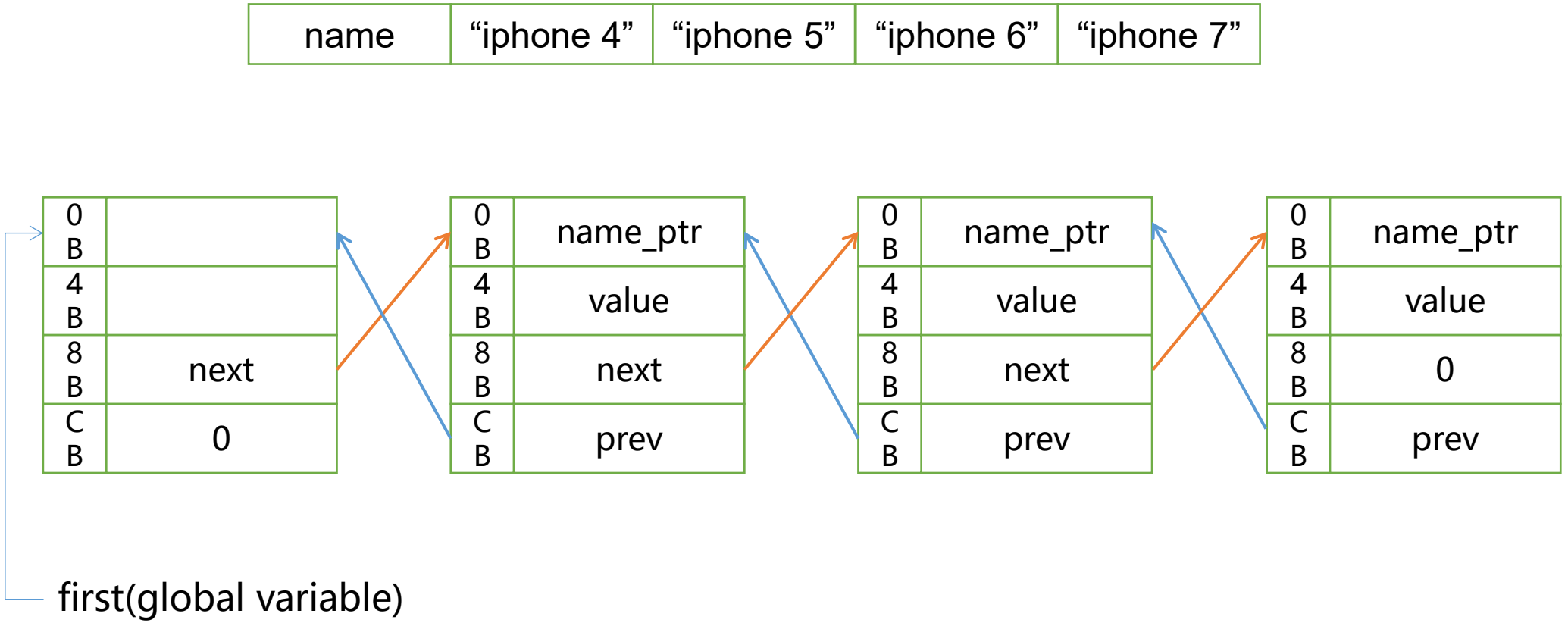
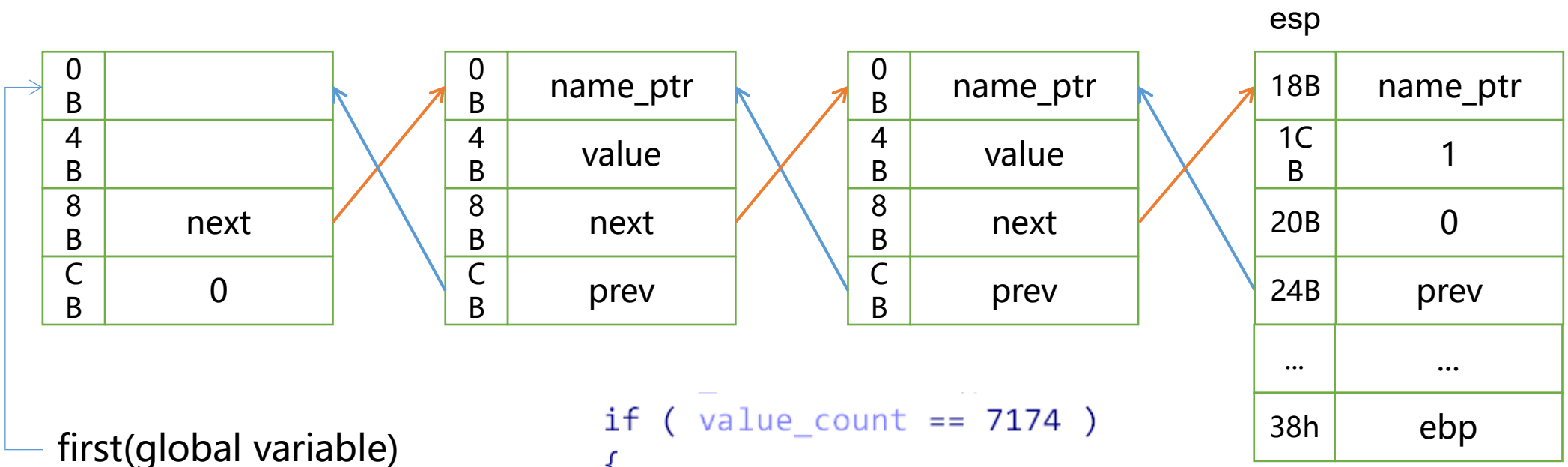
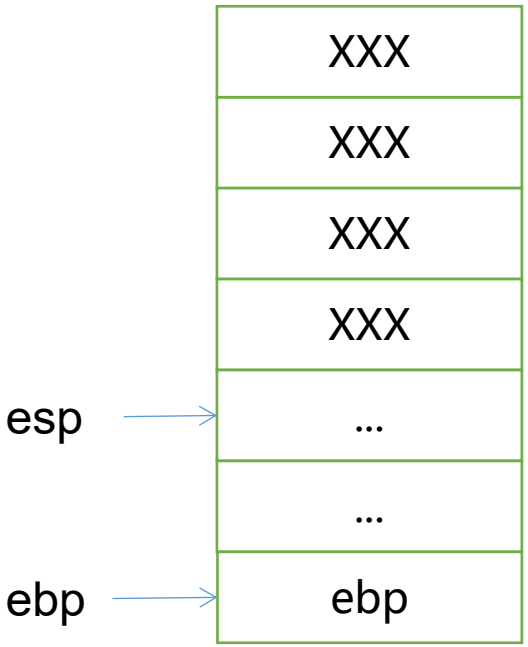

name
“iphone 4”
“iphone 5”
“iphone 6”
“iphone 7”
0 B
4 B
8 B
C B
next
0
name_ptr
value
prev
first(global variable)

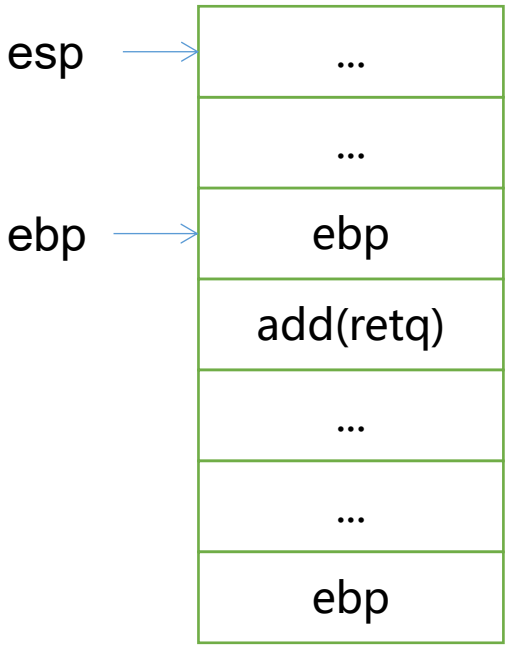
name	“iphone 4”	“iphone 5”	“iphone 6”	“iphone 7”



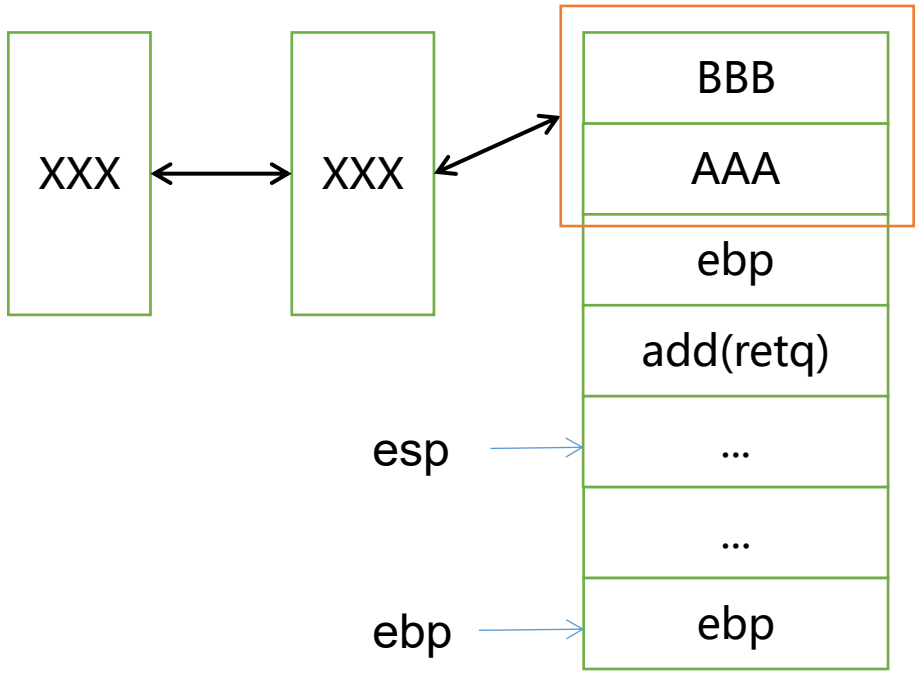

```
if ( value_count == 7174 )
{
    puts("*: iPhone 8 - $1");
    asprintf(&v2, "%s", "iPhone 8");
    v3 = 1;
    insert((int)&v2);
    value_count = 7175;
}
```



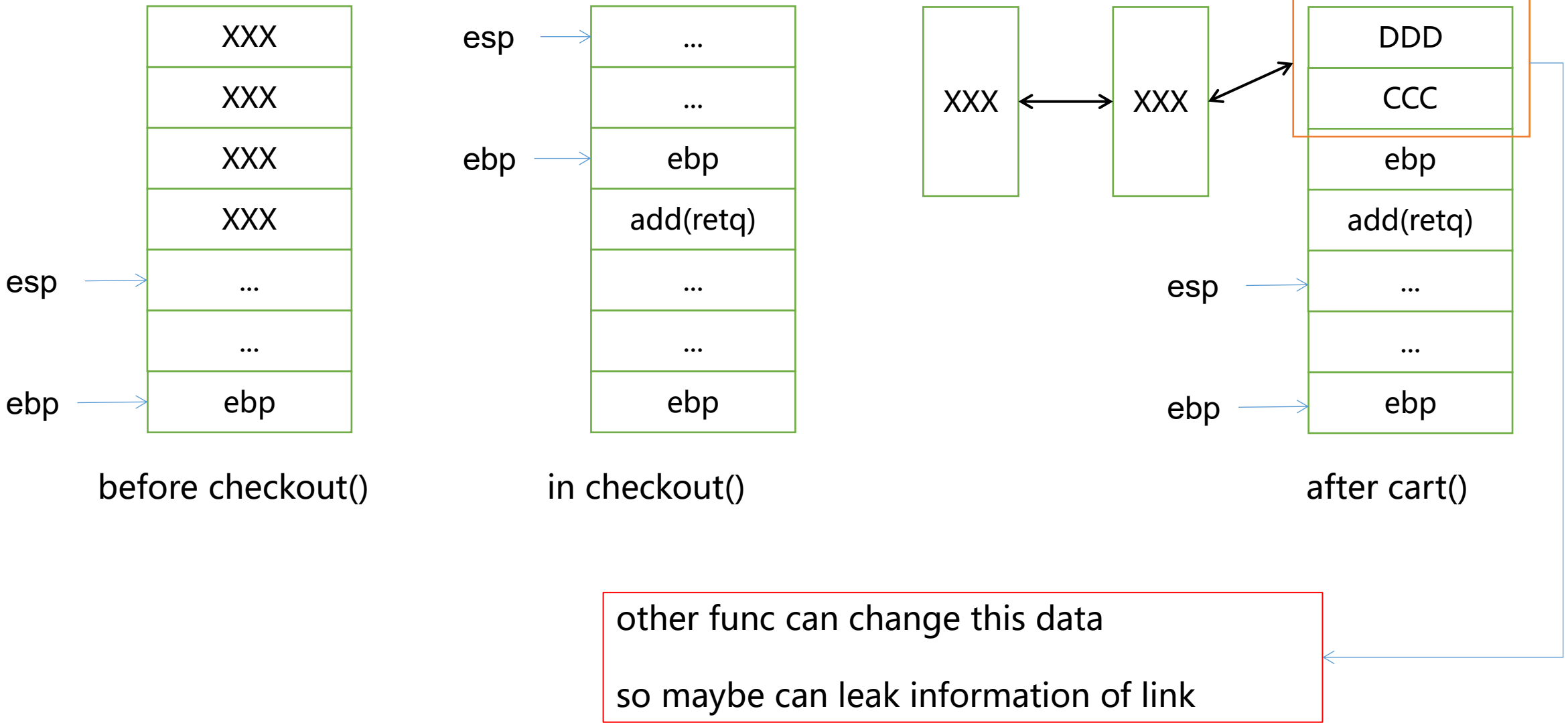
before checkout()

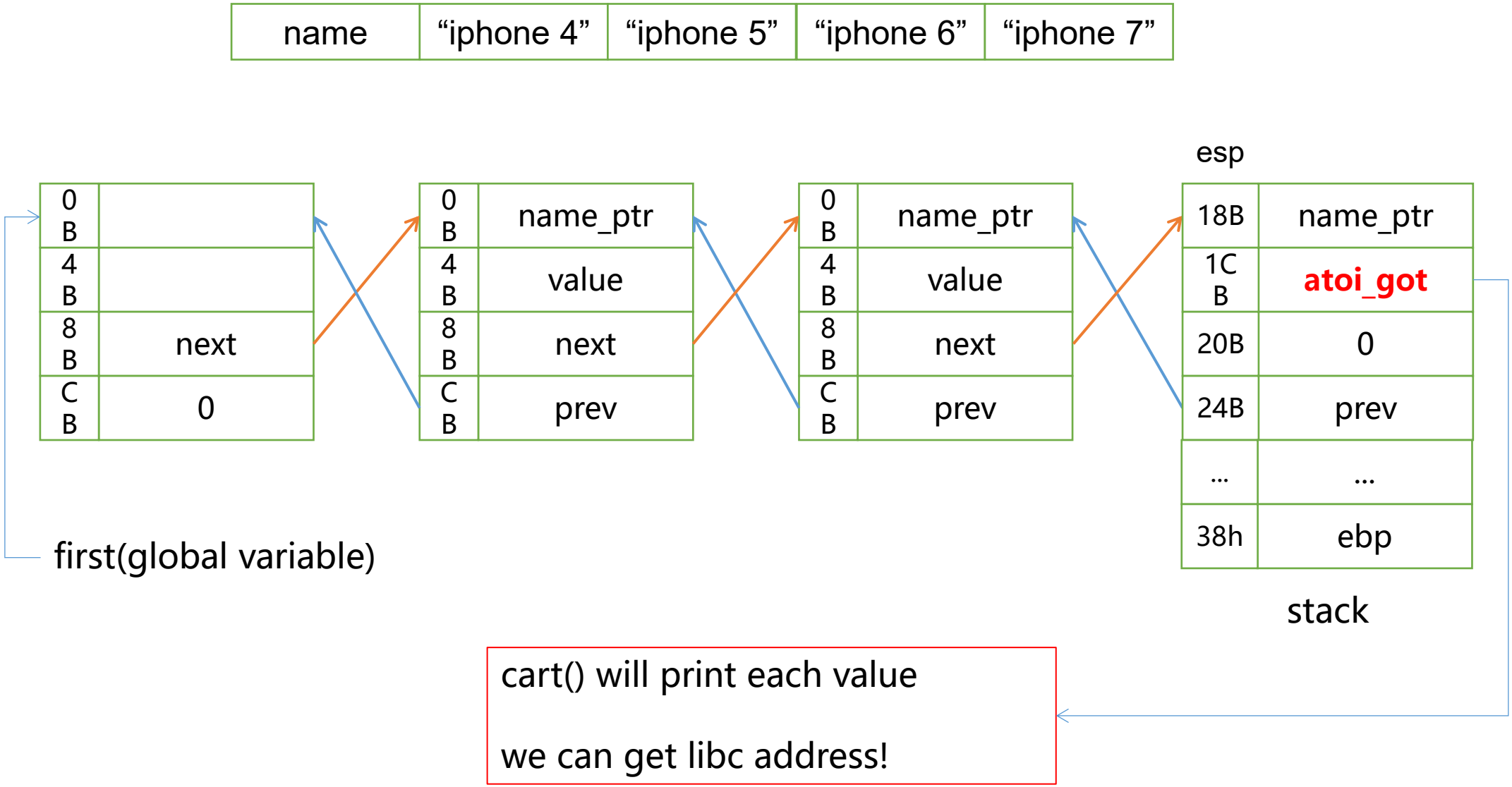


in checkout()



after checkout()
before cart()


XXX
XXX
XXX
XXX
esp
...
...
ebp
ebp
before checkout()
esp
...
...
ebp
add(retq)
...
...
ebp
in checkout()
XXX
XXX
DDD
CCC
ebp
add(retq)
esp
...
...
ebp
after cart()
other func can change this data
so maybe can leak information of link


name
“iphone 4”
“iphone 5”
“iphone 6”
“iphone 7”
esp
0 B
4 B
8 B
C B
next
0
name_ptr
value
prev
18B
1C B
20B
24B
atoi_got
...
38h
ebp
first(global variable)
stack
cart() will print each value
we can get libc address!

next? emmm...double link? Wonderful! Unlink may be sucessful!

esp

18B	name_ptr
1CB	1
20B	0
24B	prev
...	...
38h	ebp

stack

esp

18B	name_ptr
1CB	1
20B	**atoi_got - 8**
24B	**system_addr**
...	...
38h	ebp

stack

call delete

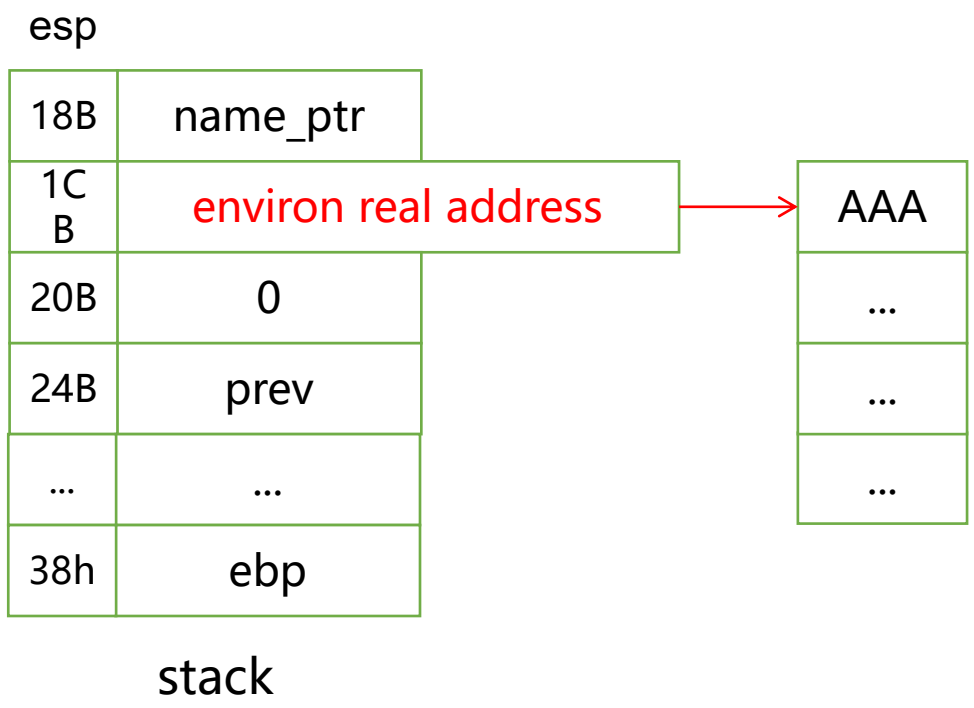
(cur->prev)->next = cur->next
(cur->next)->prev = cur->prev

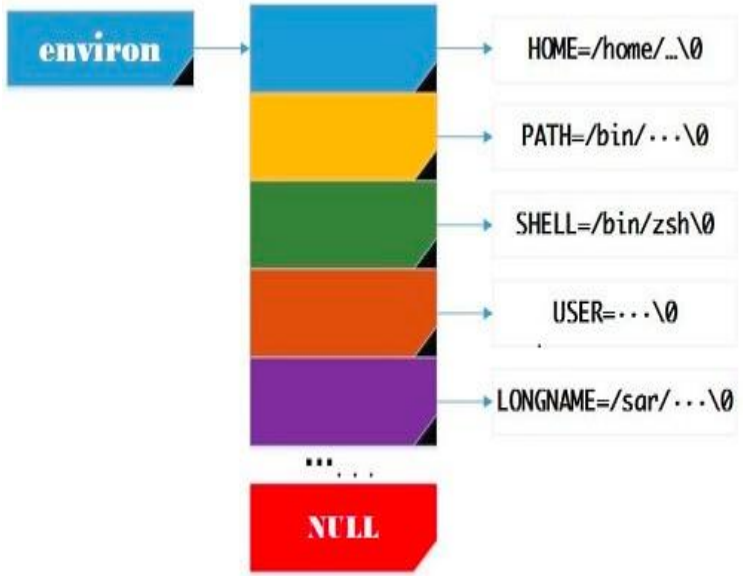
(atoi_got-8)+8 = system_addr
(sys_addr)+12 = atoi_got

atoi_got = system_addr
sys_add + 12 = atoi_got

system_addr + 12 : Unwrittable!!!!

Unlink not work, **using environ**





Unlink not work, **using environ**

esp

18B	name_ptr
1CB	environ real address
20B	0
24B	prev
...	...
38h	ebp

stack

AAA
...
...
...

stack

...	...
...	...
BBB	old ebp
...	retq
...	...
...	...
AAA	...
...	...

1. cal() print AAA
2. using gdb get (BBB-AAA)
3. we can get BBB, that is cal() stack base
4. by assembly code, we can get all function stack base

What we can do is just unlink...

esp

18B	name_ptr
1CB	1
20B	0
24B	prev
...	...
38h	ebp

stack

esp

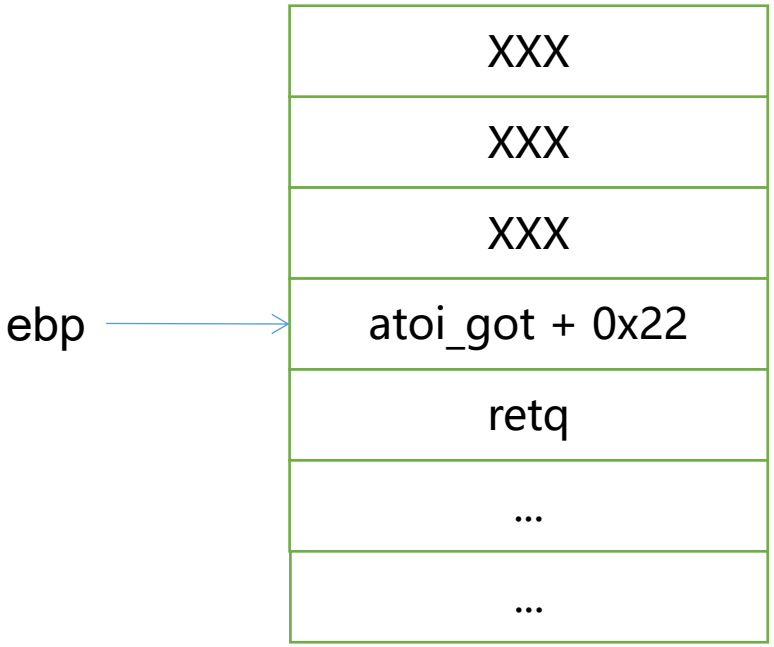
18B	name_ptr
1CB	1
20B	**atoi_got + 0x22**
24B	**del_ebp - 8**
...	...
38h	ebp

stack

call delete

(atoi_got + 0x22) + 12 = del_ebp - 8
(del_ebp - 8) + 8 = atoi_got + 0x22

atoi_got + 0x2C = del_ebp - 8
del_ebp = atoi_got + 0x22

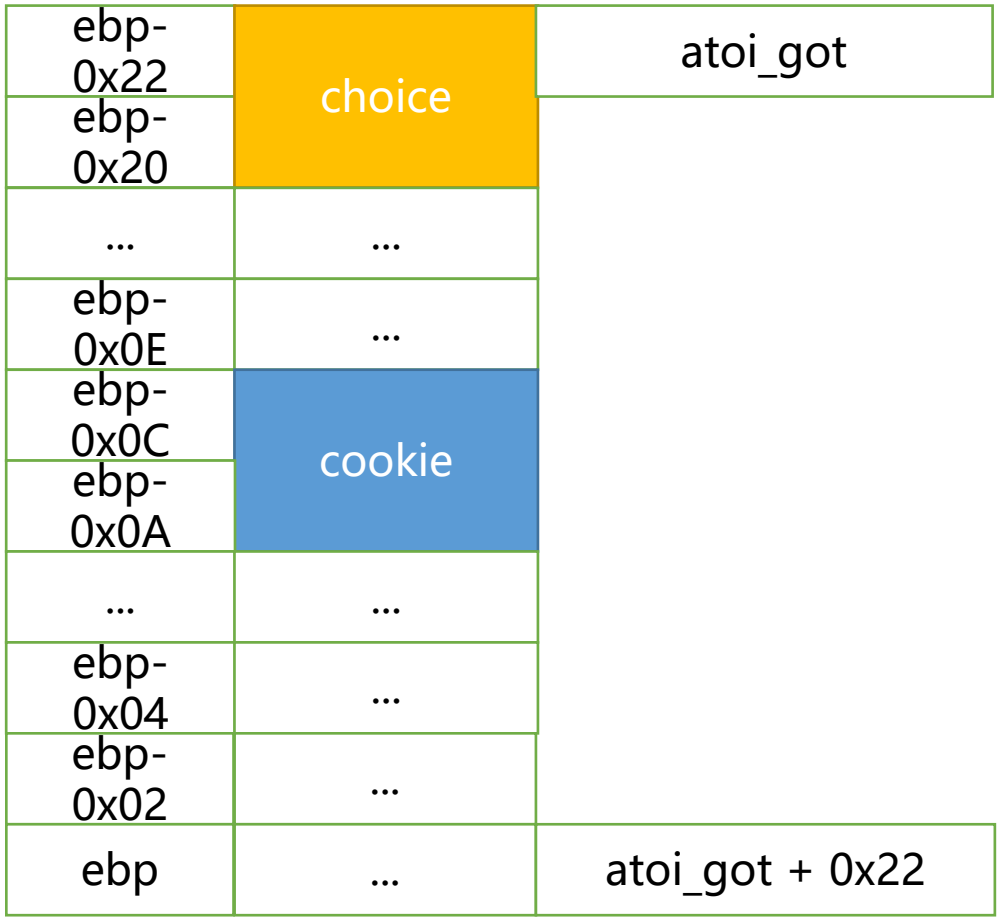


leave ret(del func)

下面回到主函数继续输入 choice

my_read(&choice, 0x15u)

switch(atoi(&choice))

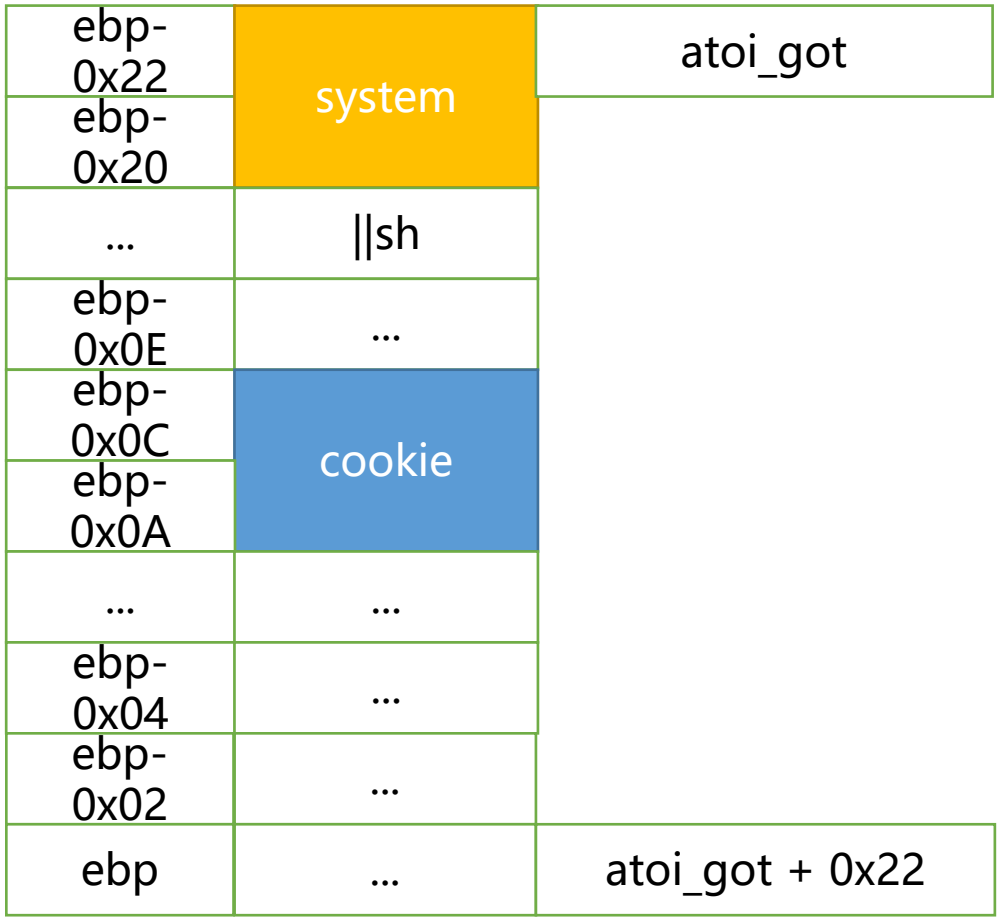


leave ret(del func)

下面回到主函数继续输入choice

my_read(&choice, 0x15u)

switch(atoi(&choice))

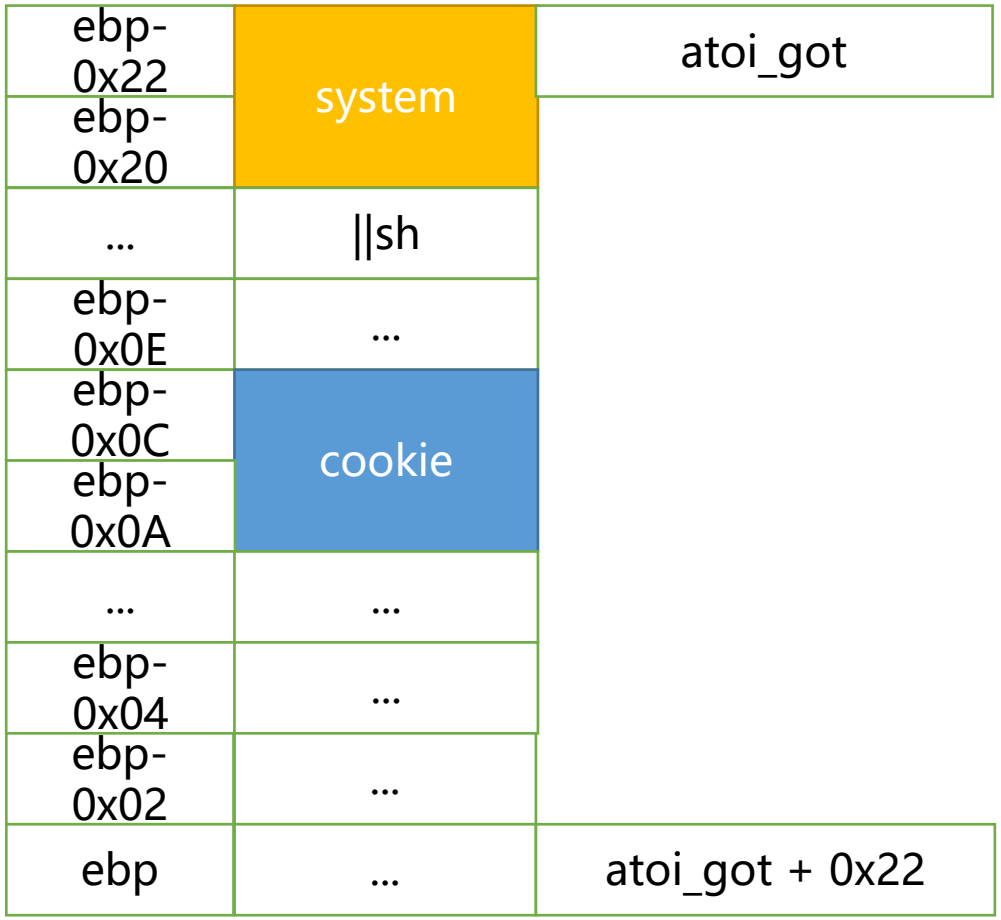


leave ret(del func)

下面回到主函数继续输入choice

my_read(&choice, 0x15u)

switch(atoi(&choice))



leave ret(del func)

下面回到主函数继续输入choice

my_read(&choice, 0x15u)

switch(atoi(&choice))

相当于执行 system(“system||sh”)