

Double Free

Step.1

`p0 = malloc(504)`

`p1 = malloc(512)`

chunk 0

512 + 1

p0

[504]

chunk 1

520 + 1

p1

[512]

break

un_size + 1

unallocated



Step.2
free(p0)

chunk 1

512 + 1

fd

bk

512

520 + 0

[512]

un_size + 1

unallocated

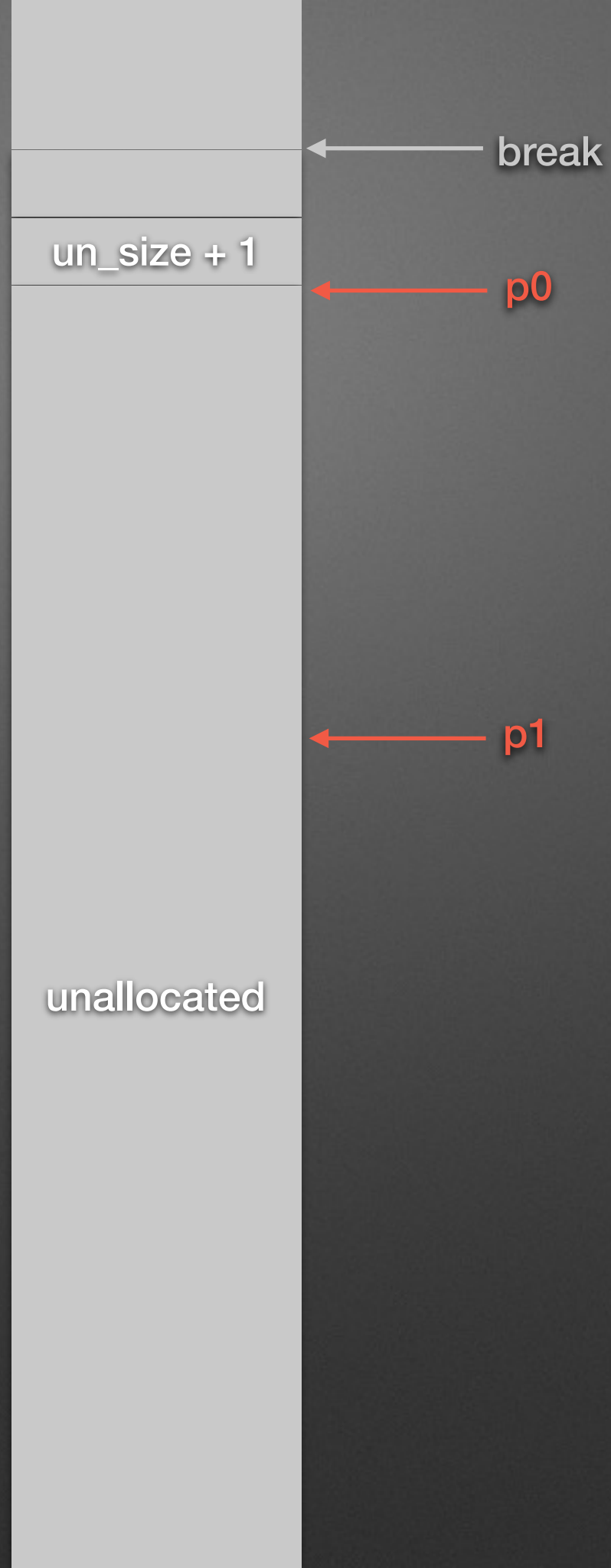
p0

p1

break



Step.3
free(p1)



Step.4

`p2 = malloc(768)`

chunk 2

776 + 1

`p0, p2`

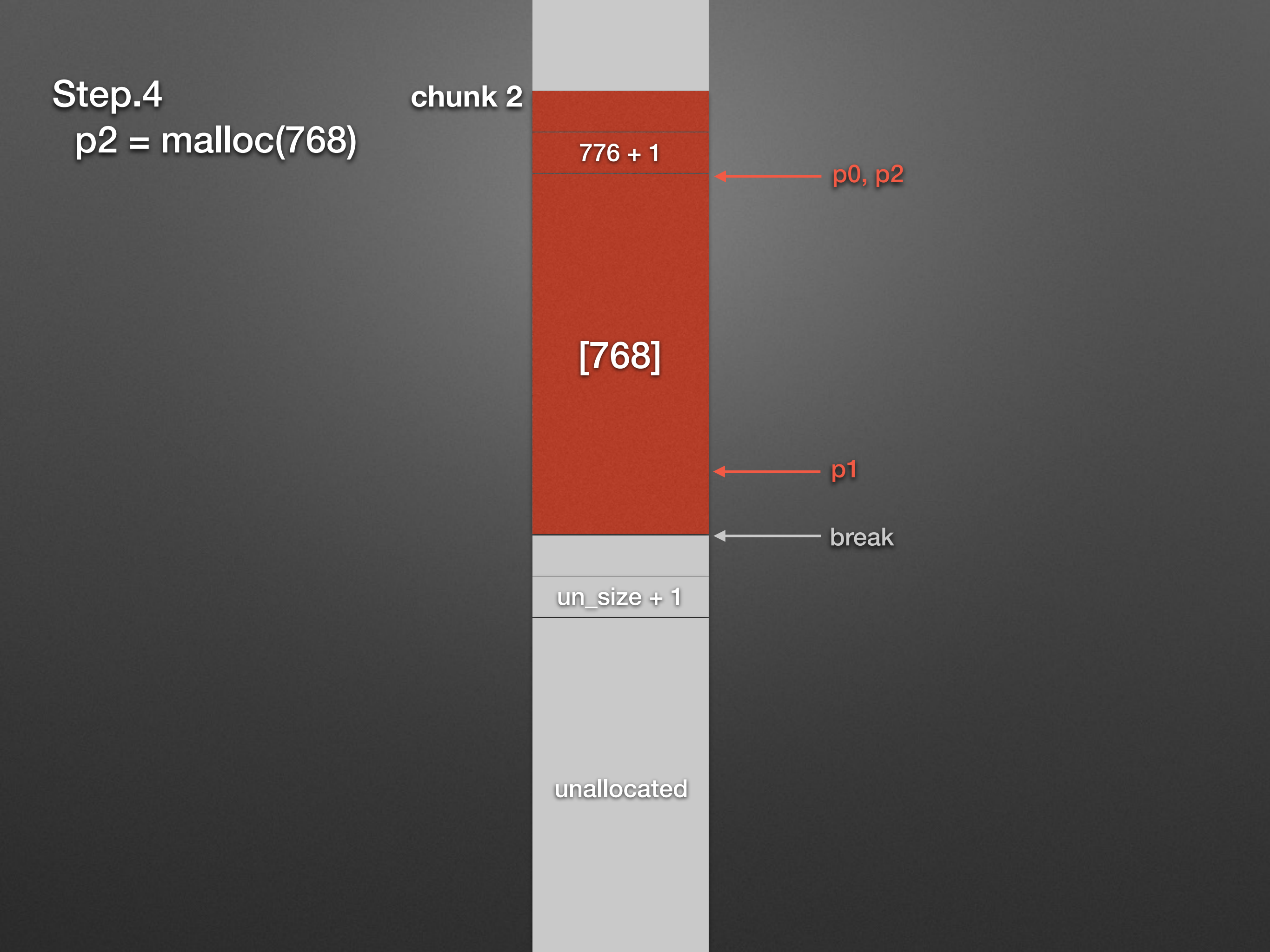
[768]

`p1`

break

un_size + 1

unallocated

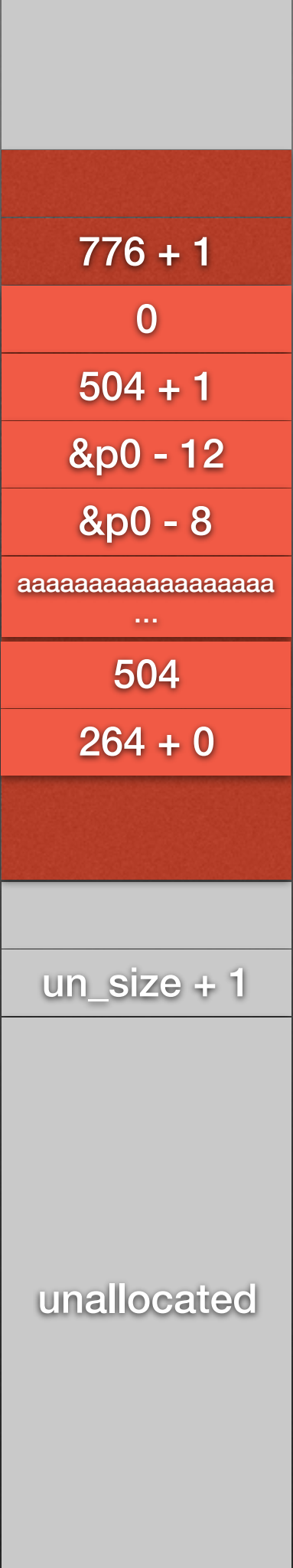


Step.5
fake chunk P0

chunk 2

fake P0

fake P1



p0, p2

p1

break

Step.6

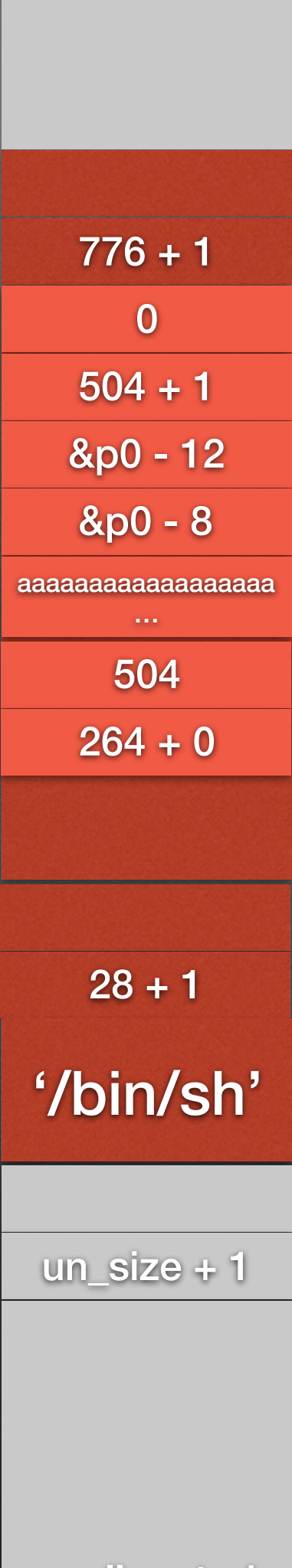
```
p3 = malloc(20)
# '/bin/sh'
```

chunk 2

fake P0

fake P1

chunk 3



p0, p2

p1

p3

break

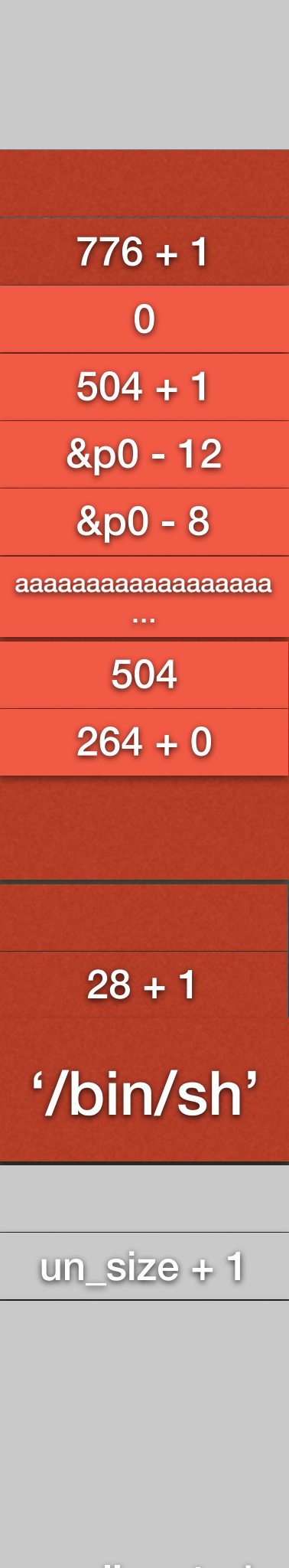
Step.7
Double Free
free(p1)

chunk 2

fake P0

fake P1

chunk 3



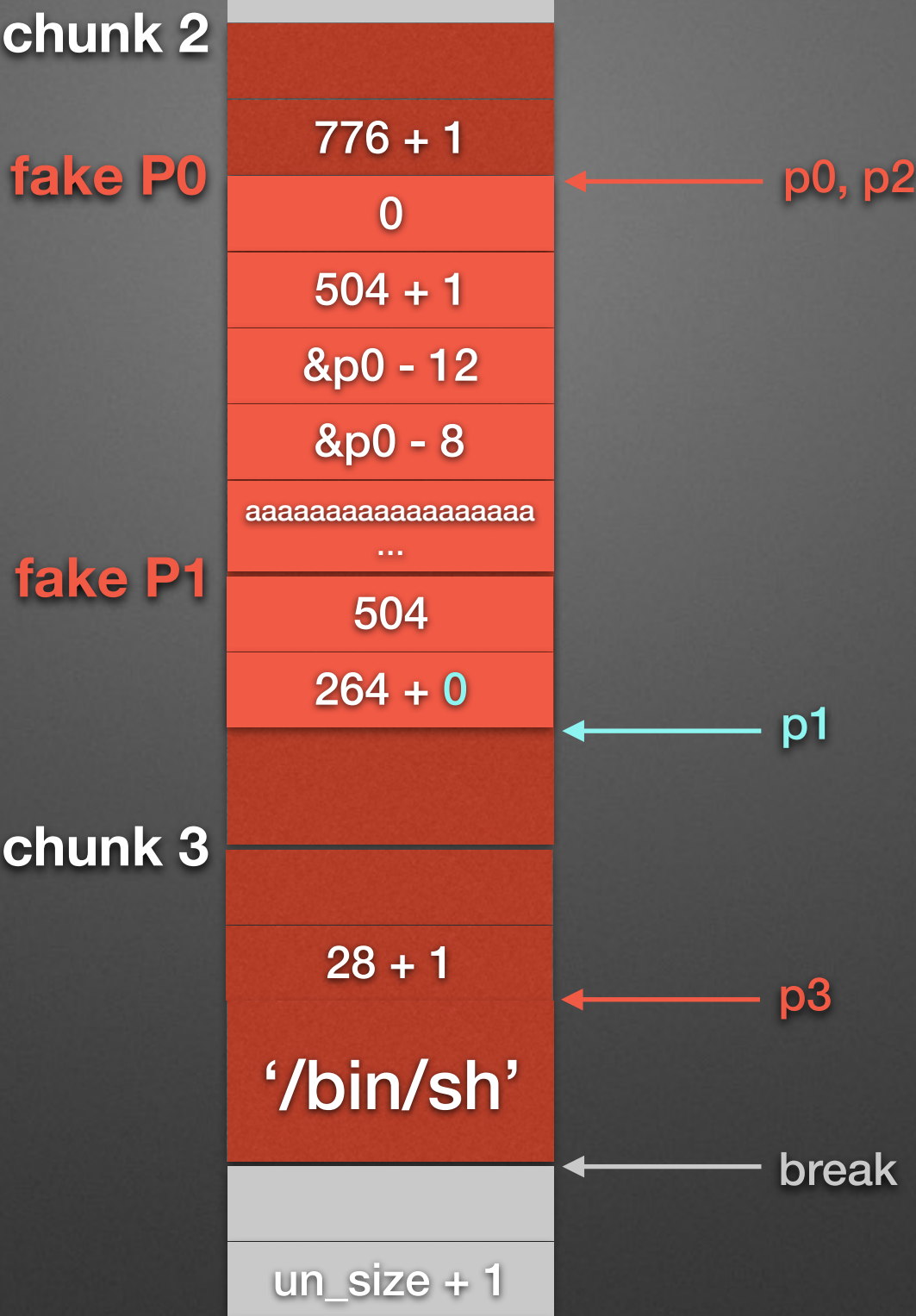
p0, p2

p1

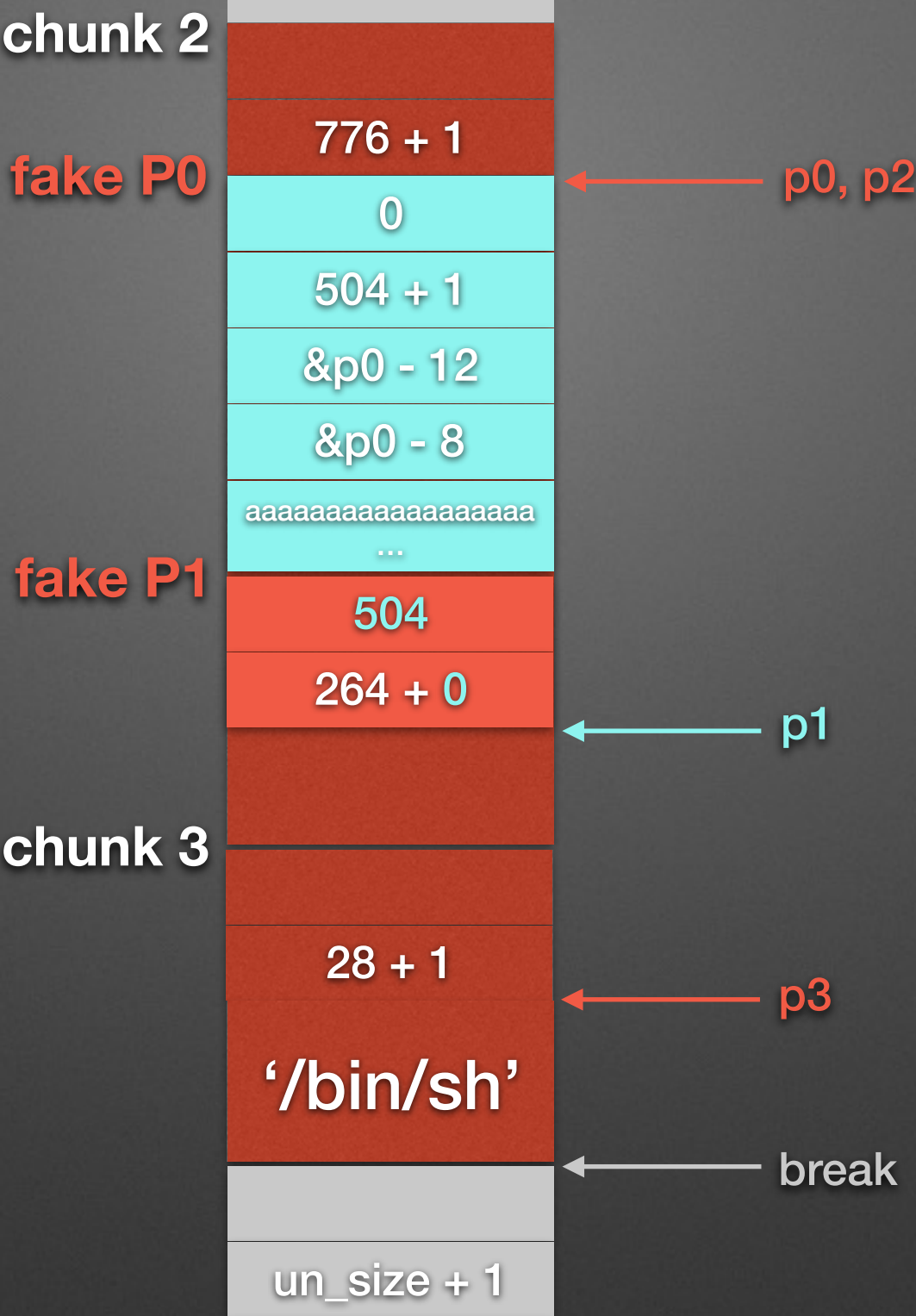
p3

break

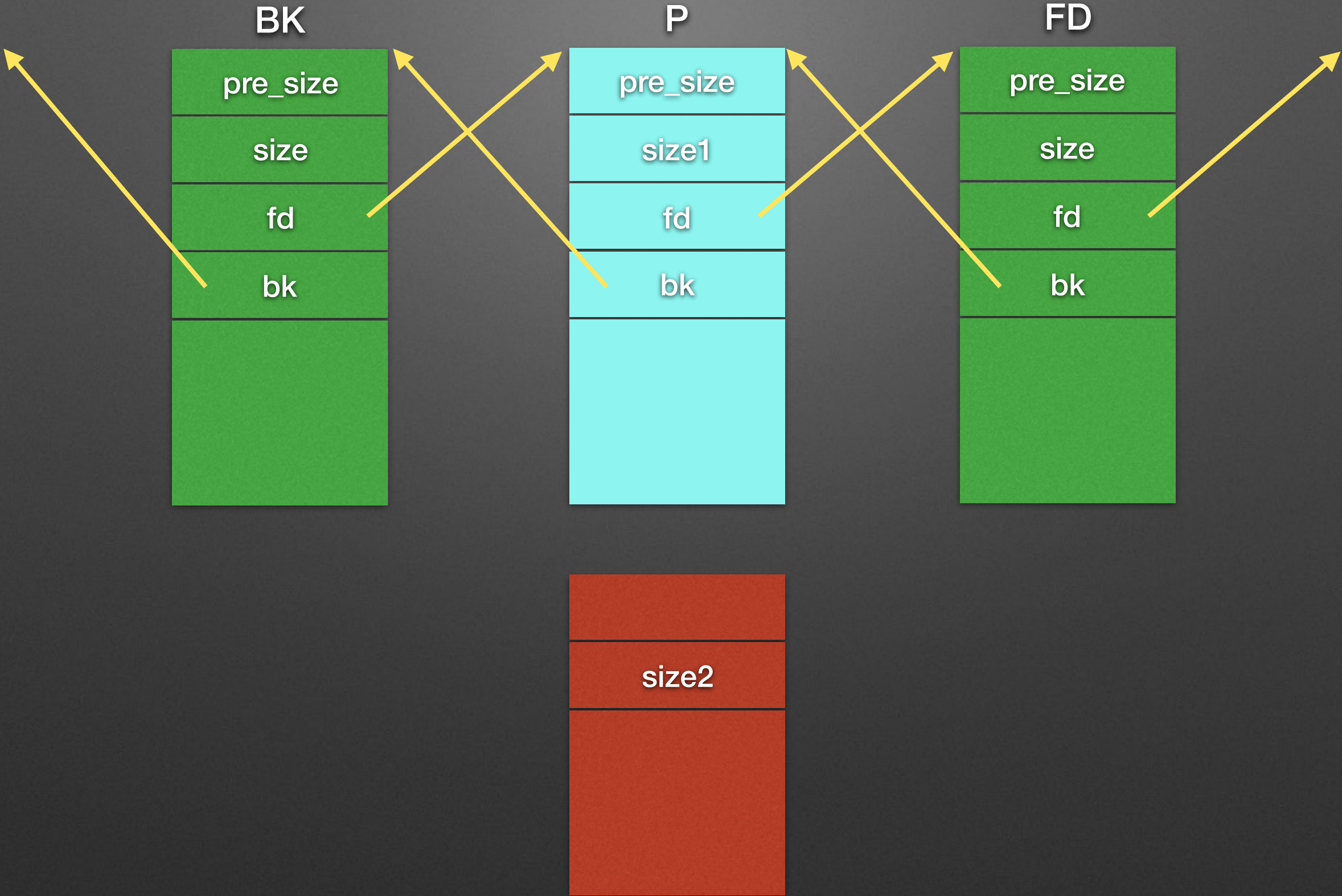
Step.7
Double Free
free(p1)



Step.7
Double Free
free(p1)



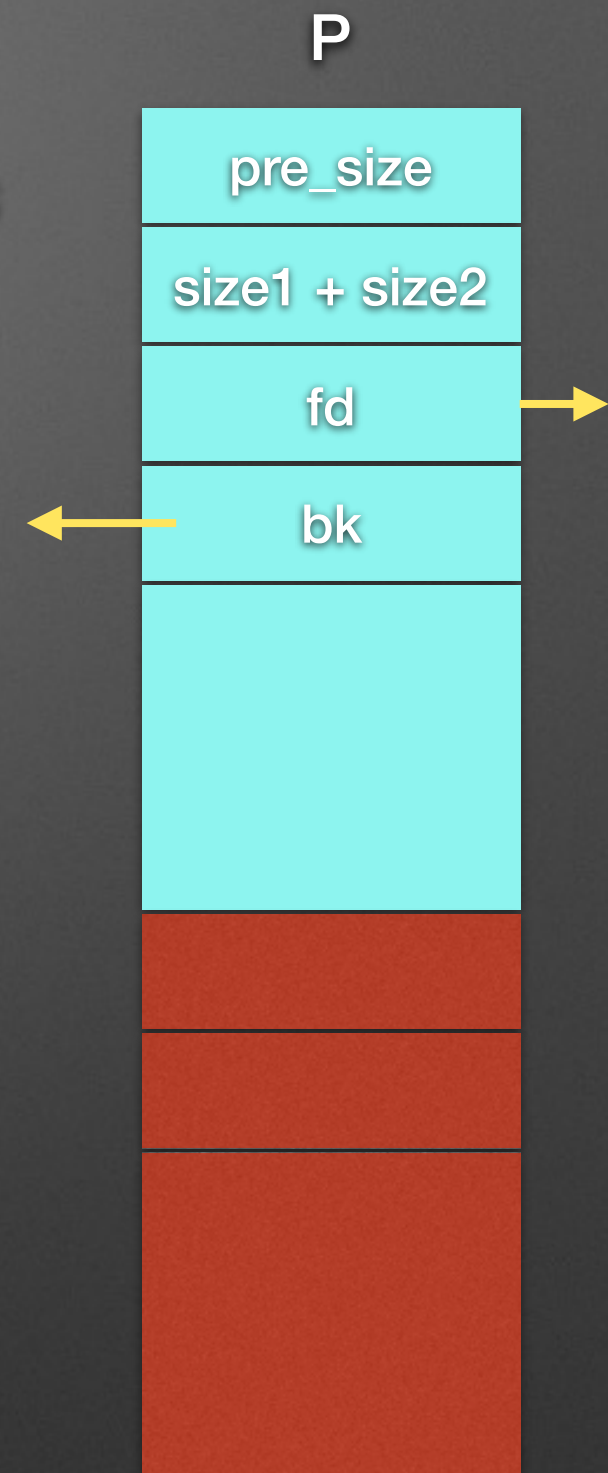
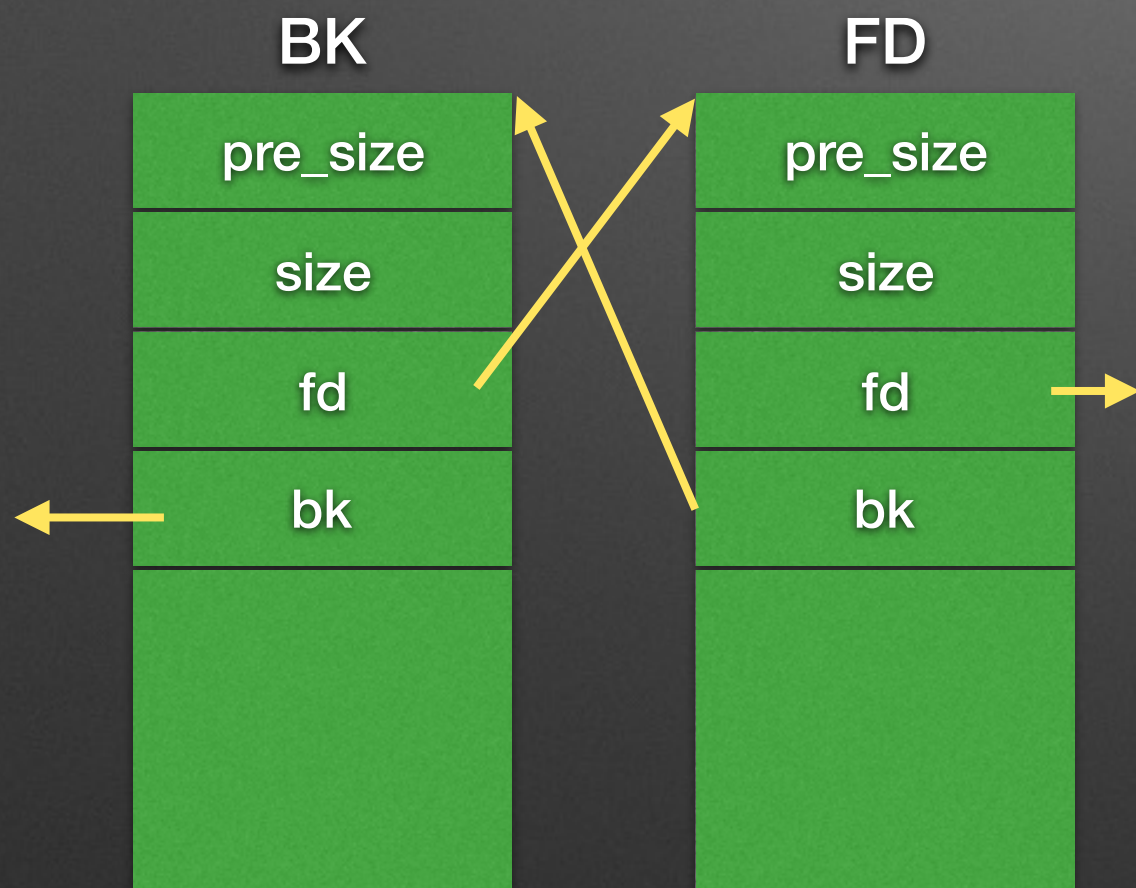
空闲堆块合并



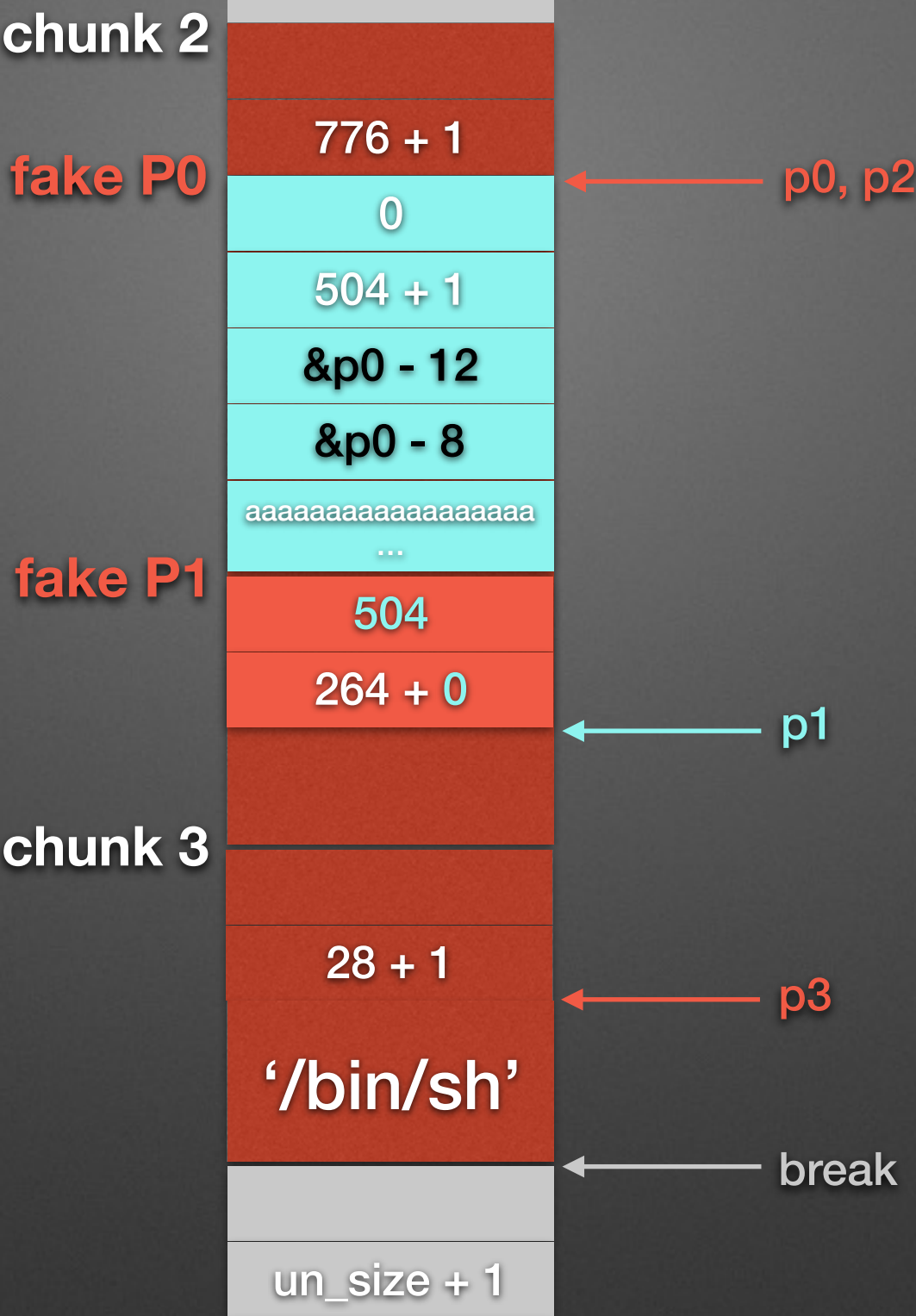

```

FD = P->fd;
BK = P->bk;
if (__builtin_expect (FD->bk != P || BK->fd != P, 0))
    malloc_printerr (check_action, "corrupted double-linked list", P, AV);
else {
    FD->bk = BK;
    BK->fd = FD;
    ...
}

```



Step.7
Double Free
free(p1)



[heap]

[.data]

chunk 2

fake P0

776 + 1

0

504 + 1

&p0 - 12

&p0 - 8

aaaaaaaaaaaaaaaaaaaa

...

fake P1

504

264 + 0

p1

chunk 3

28 + 1

‘/bin/sh’

p3

break

un_size + 1

&p0 = 0x0804bfa0

p0

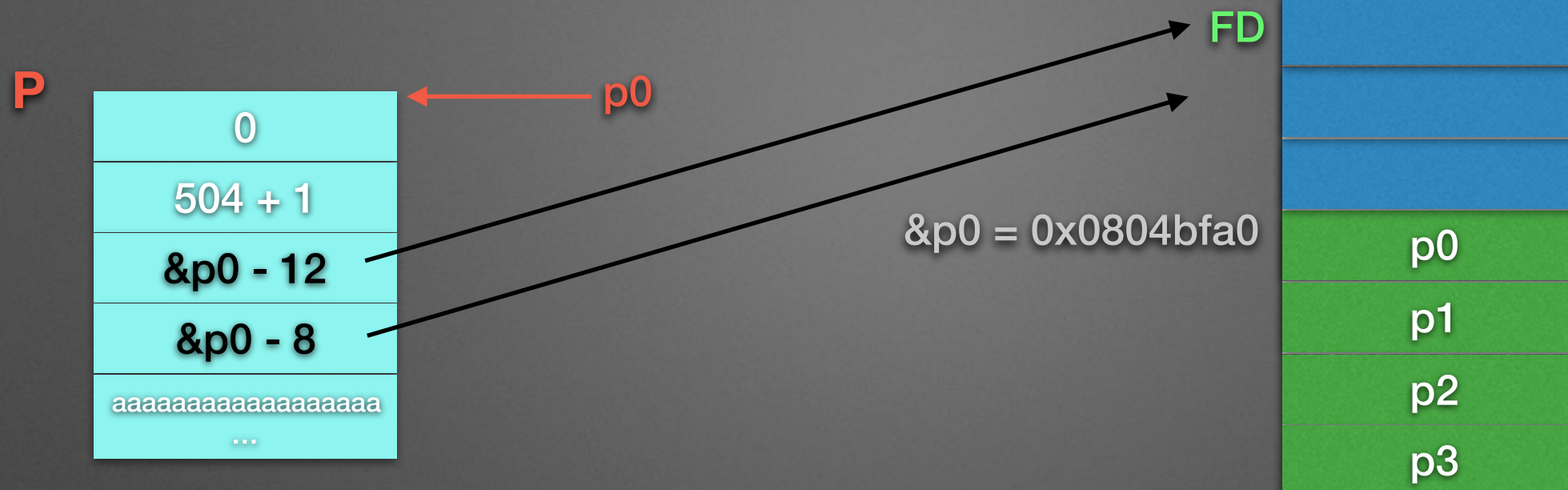
p1

p2

p3

[heap]

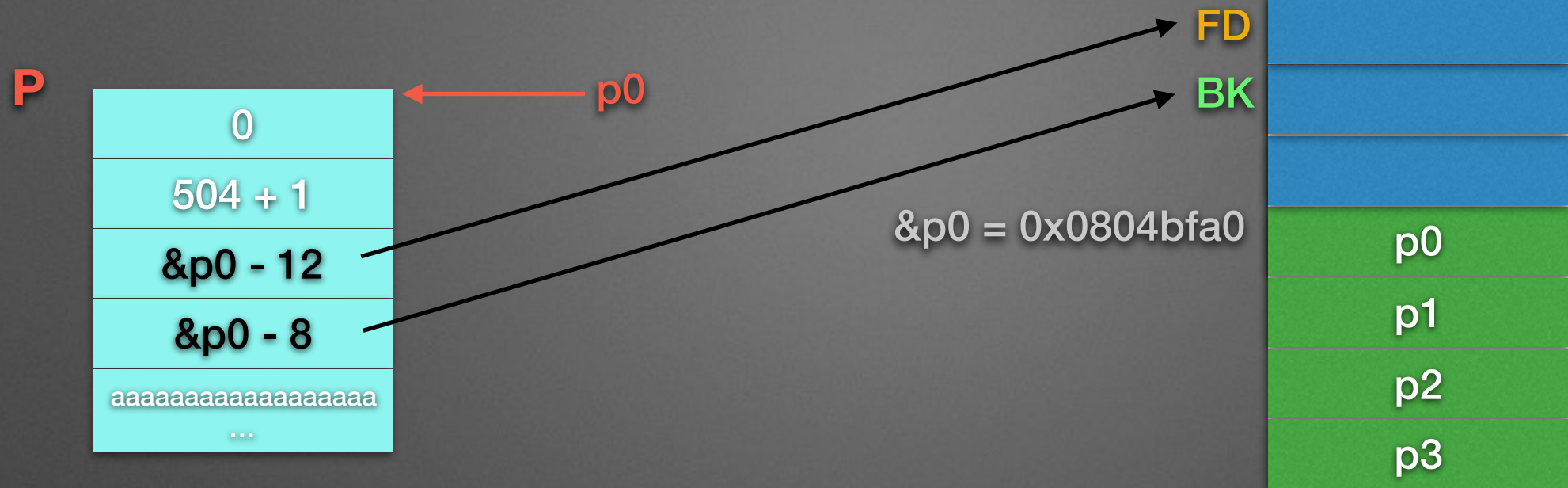
[.data]



```
FD = P->fd;
BK = P->bk;
if (__builtin_expect (FD->bk != P || BK->fd != P, 0))
    malloc_printerr (check_action, "corrupted double-linked list", P, AV);
else {
    FD->bk = BK;
    BK->fd = FD;
    ...
}
```

[heap]

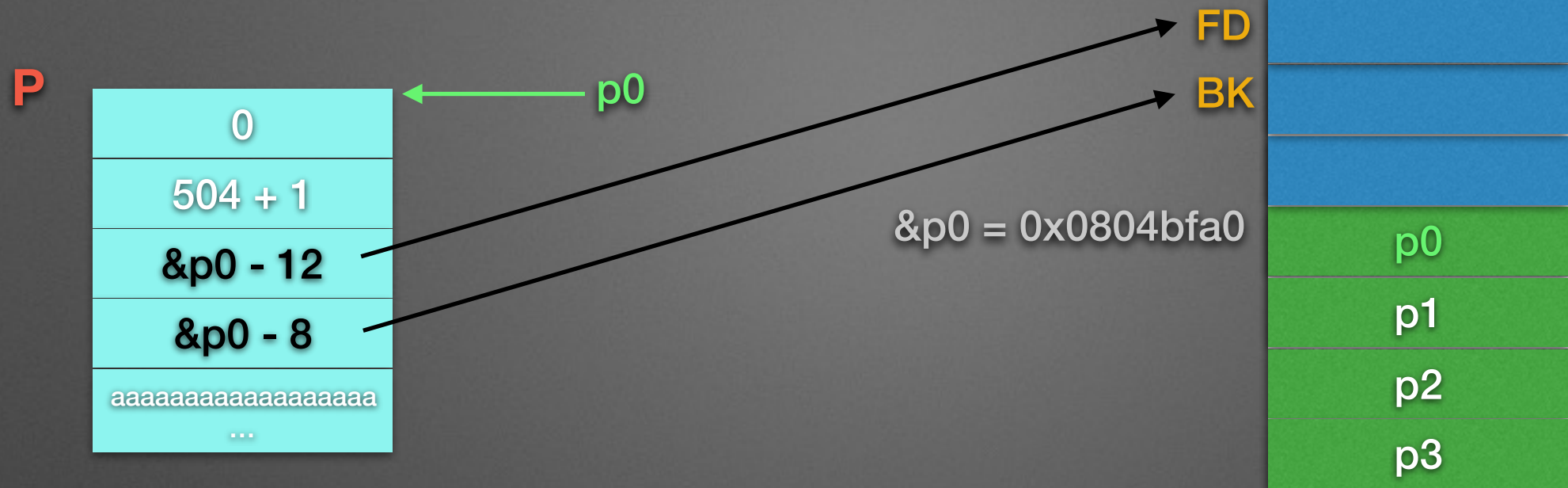
[.data]



```
FD = P->fd;
BK = P->bk;
if (__builtin_expect (FD->bk != P || BK->fd != P, 0))
    malloc_printerr (check_action, "corrupted double-linked list", P, AV);
else {
    FD->bk = BK;
    BK->fd = FD;
    ...
}
```

[heap]

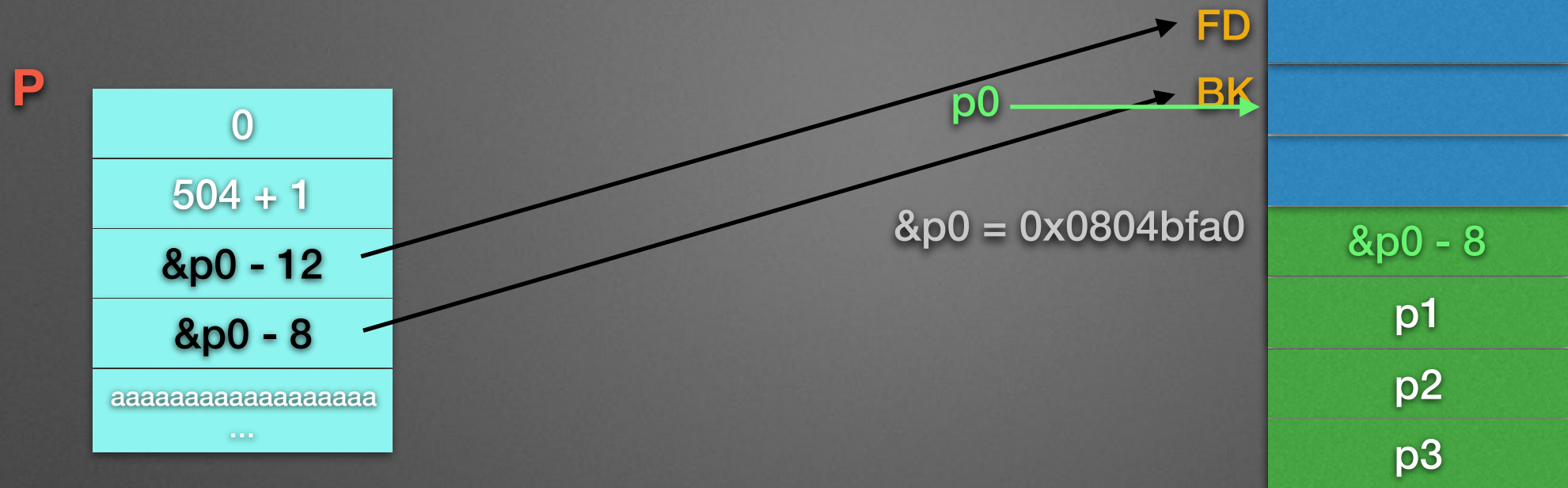
[.data]



```
FD = P->fd;
BK = P->bk;
if (__builtin_expect (FD->bk != P || BK->fd != P, 0))
    malloc_printerr (check_action, "corrupted double-linked list", P, AV);
else {
    FD->bk = BK;
    BK->fd = FD;
    ...
}
```


[heap]

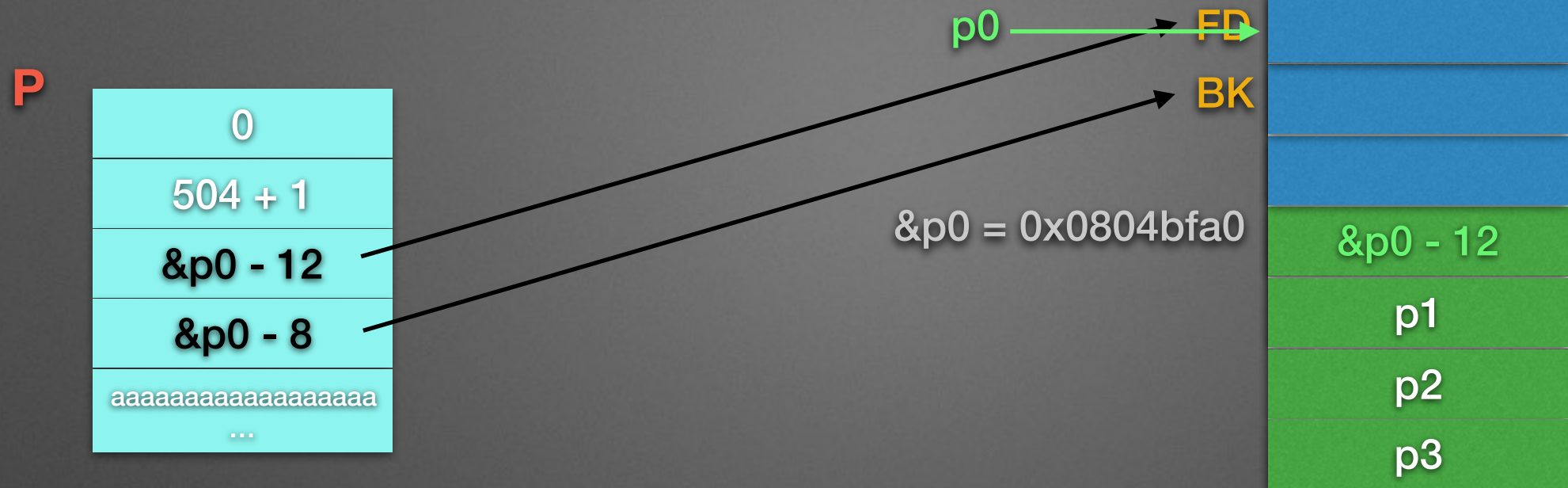
[.data]



```
FD = P->fd;
BK = P->bk;
if (__builtin_expect (FD->bk != P || BK->fd != P, 0))
    malloc_printerr (check_action, "corrupted double-linked list", P, AV);
else {
    FD->bk = BK;
    BK->fd = FD;
    ...
}
```

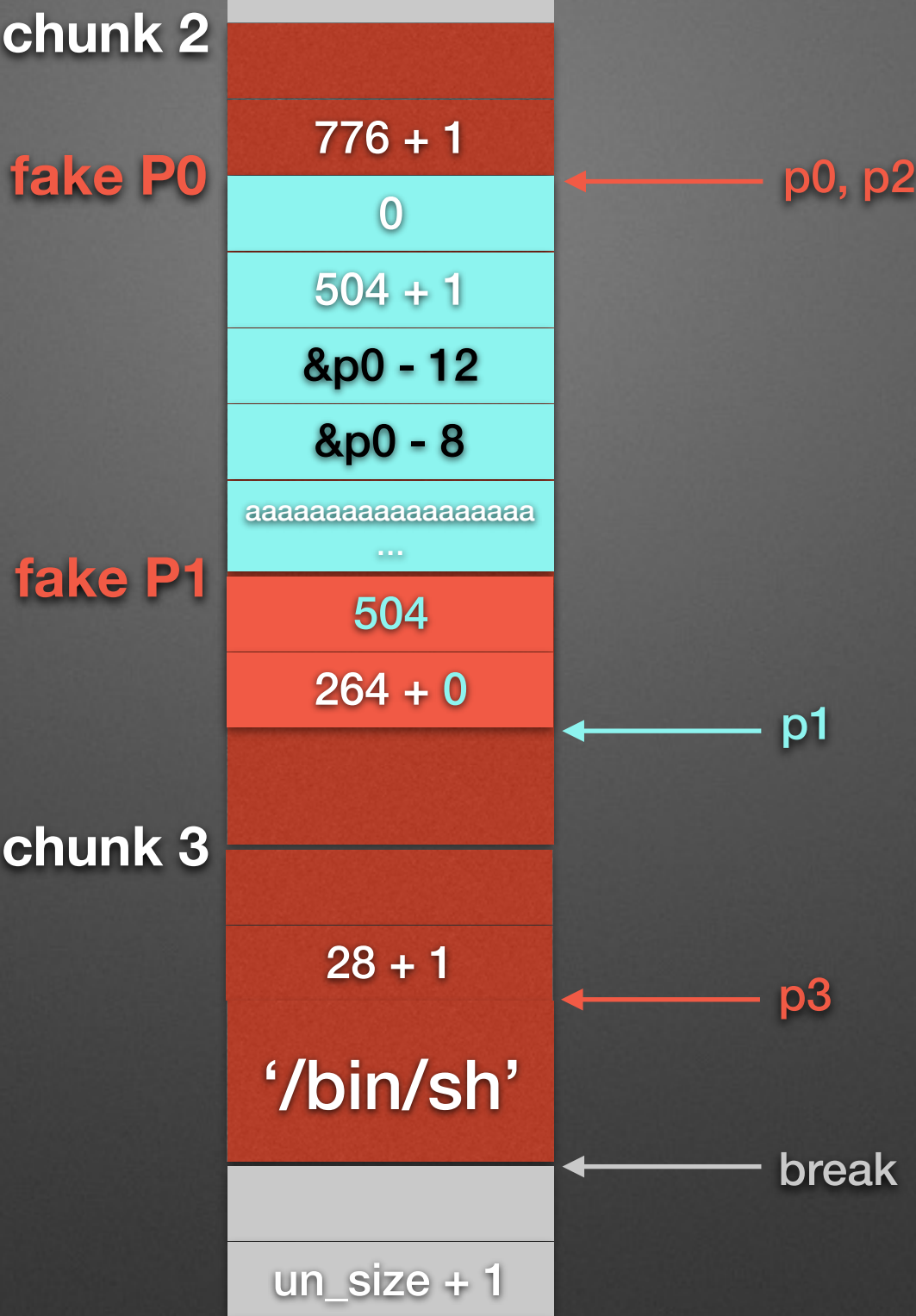
[heap]

[.data]



```
FD = P->fd;
BK = P->bk;
if (__builtin_expect (FD->bk != P || BK->fd != P, 0))
    malloc_printerr (check_action, "corrupted double-linked list", P, AV);
else {
    FD->bk = BK;
    BK->fd = FD;
    ...
}
```

Step.7
Double Free
free(p1)



Step.7
Double Free
free(p1)

chunk 2

fake P0

chunk 3

776 + 1

0

768 + 1

fd

bk

28 + 1

‘/bin/sh’

un_size + 1

p0

p2

p1

p3

break

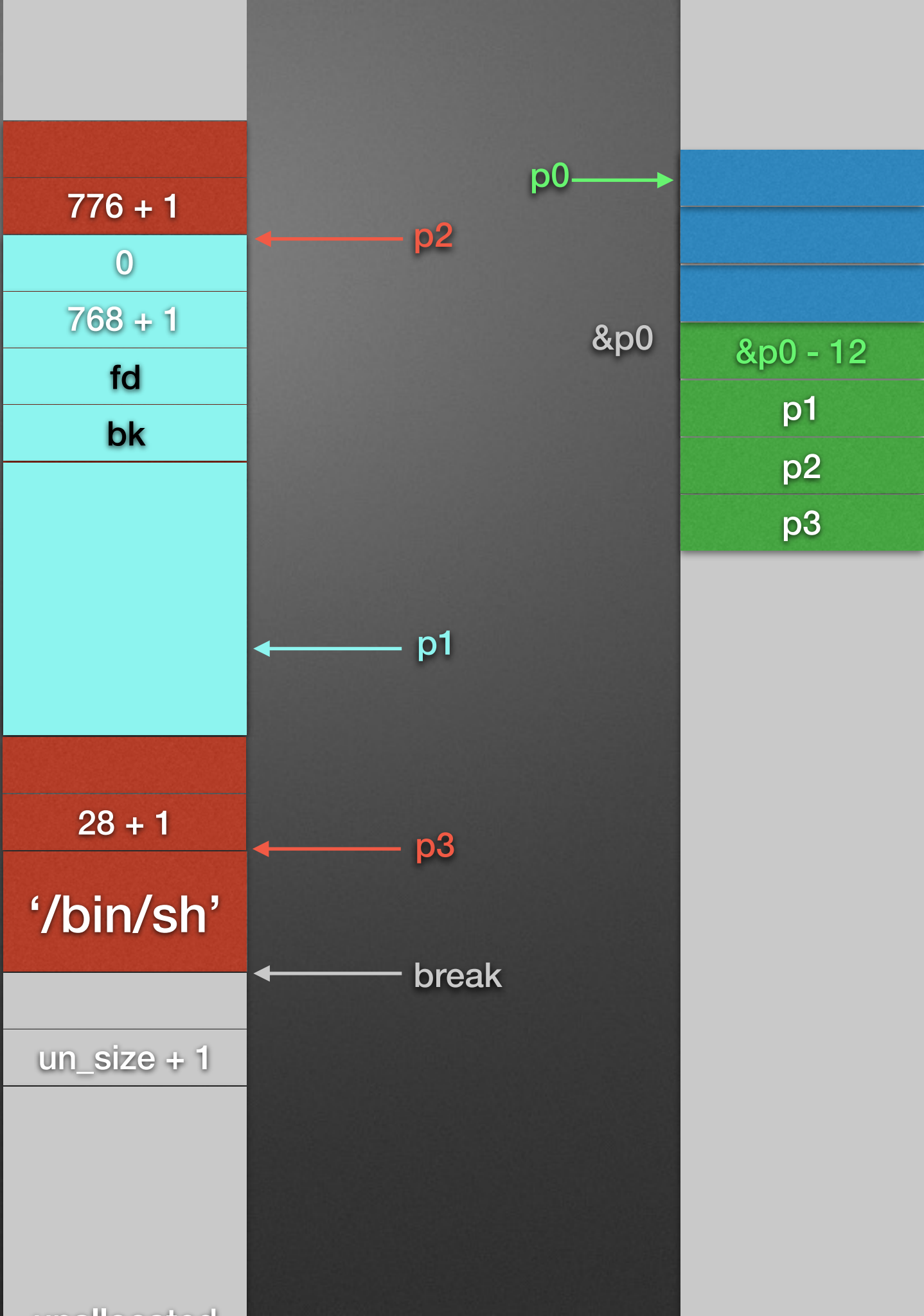
&p0

&p0 - 12

p1

p2

p3



Step.8
edit(p0)

chunk 2

fake P0

chunk 3

776 + 1

0

768 + 1

fd

bk

28 + 1

‘/bin/sh’

un_size + 1

got_addr

free@got

p0

&p0

&p0 - 12

p1

p2

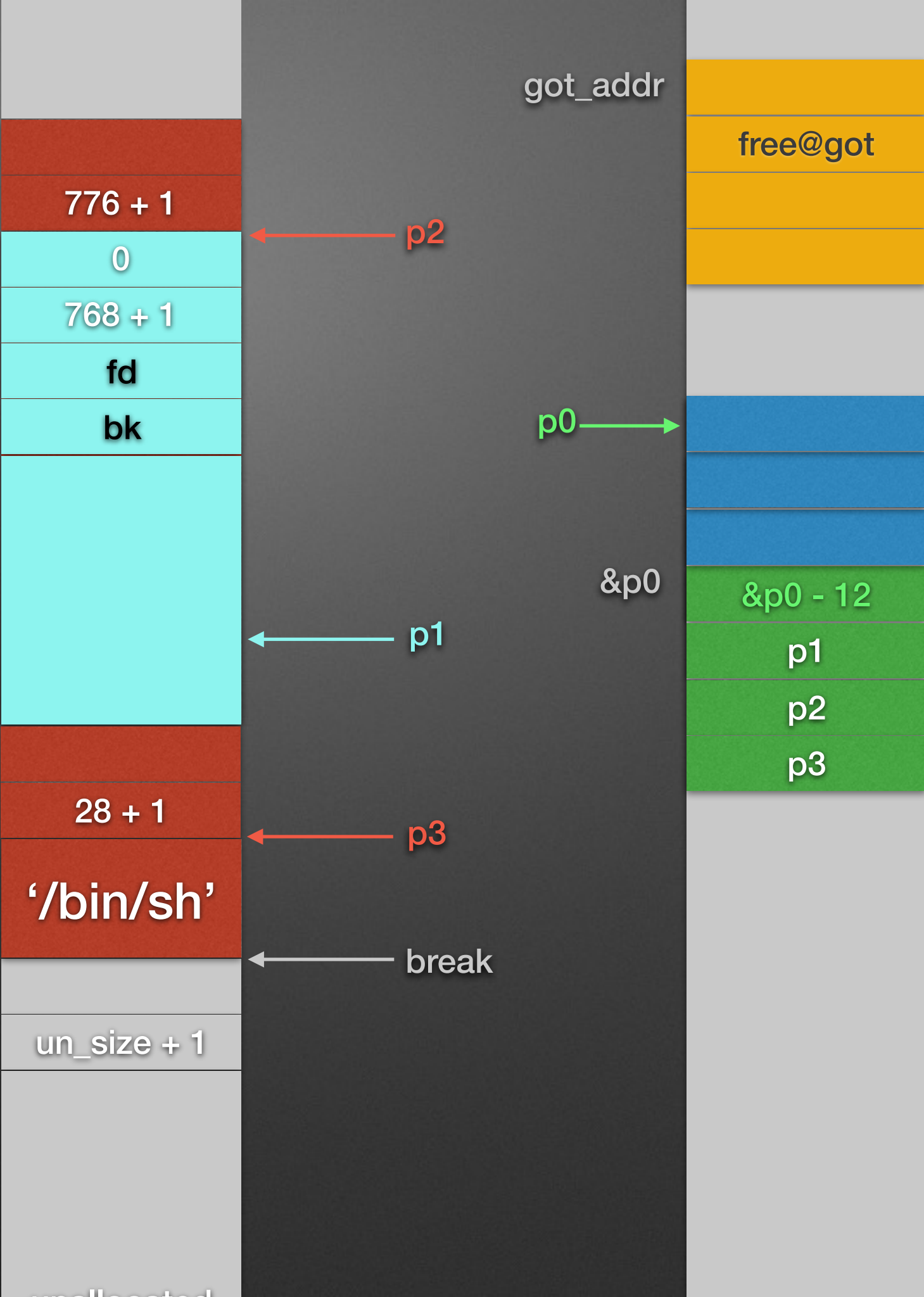
p3

p2

p1

p3

break



Step.8
edit(p0)
'a'*12 + got_addr

chunk 2

fake P0

chunk 3

got_addr

free@got

776 + 1

0

768 + 1

fd

bk

p0

'aaaa'

'aaaa'

'aaaa'

&p0

got_addr

p1

p2

p3

p2

p1

p3

break

un_size + 1

unallocated

Step.8
edit(p0)
'a'*12 + got_addr

chunk 2

fake P0

chunk 3

p0 → got_addr

free@got

776 + 1

0

768 + 1

fd

bk

p2

'aaaa'

'aaaa'

'aaaa'

&p0

got_addr

p1

p2

p3

p1

28 + 1

p3

'/bin/sh'

break

un_size + 1

Step.9
print(p0)
leak GOT

chunk 2

fake P0

chunk 3

p0 → got_addr

free@got

776 + 1

0

768 + 1

fd

bk

'aaaa'

'aaaa'

'aaaa'

&p0

got_addr

p1

p2

p3

p2

p1

p3

break

un_size + 1

unallocated

Step.10

edit(p0)

free@got -> system

chunk 2

fake P0

chunk 3

p0 -> got_addr

p2

p1

p3

break

&p0

system_addr

'aaaa'

'aaaa'

'aaaa'

got_addr

p1

p2

p3

776 + 1

0

768 + 1

fd

bk

28 + 1

'/bin/sh'

un_size + 1

Step.11
free(p3)
system('/bin/sh')

chunk 2

fake P0

chunk 3

p0 → got_addr

p2

p1

p3

break

&p0

system_addr

'aaaa'

'aaaa'

'aaaa'

got_addr

p1

p2

p3

776 + 1

0

768 + 1

fd

bk

28 + 1

'/bin/sh'

un_size + 1