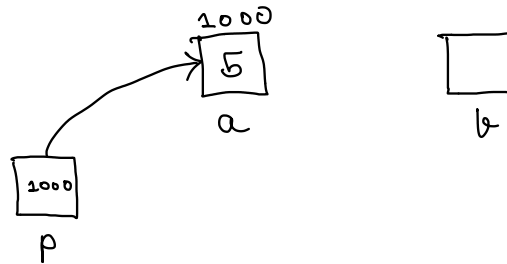


86 - Accessing Variable Through Pointer

Monday, 10 August 2026

00:11

`int a, b;` → a, and b initialised
`int *p;` → ptr p made
`p = &a` → pointer stores address of a.
`b = *p` a and *p are same now



So, $a = 5$, $\&a = 1000$
 $p = 1000$, $*p = 5$

So, `int a = 5;`
`int *p;`
`p = &a`
`printf("%d", *p);` // p has address of a
o/p: 5

∴ $b = *p$ → b gets value of a since
*p holds value of a

```
pointers/pointer1.c> int main()
```

Search...

```
1 #include <stdio.h>
2
3 int main(){
4
```

NOTE:

```

5     int a, b;
6     int c = 5;
7     int * p; // a pointer p declared
8
9     a = 4 * (c + 5);
10
11    p = &c; // pointer now holds address of c
12    // so *p automatically holds value of c
13
14    b = 4 * (*p + 5);
15
16    printf("%d\n", a);
17    printf("%d\n", b);
18
19    return 0;
20 }
21

```

$*\&c = 15$

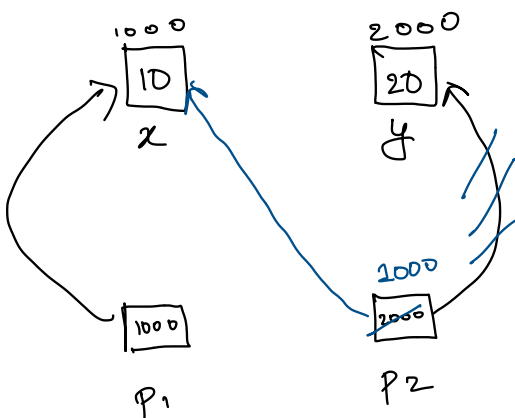
pointers — zsh

```

4040%
arnesh@Arneshs-MacBook-Air pointers % clear
arnesh@Arneshs-MacBook-Air pointers % gcc pointer1.c
arnesh@Arneshs-MacBook-Air pointers % ./a.out
40
40
arnesh@Arneshs-MacBook-Air pointers %

```

Changing Pointers

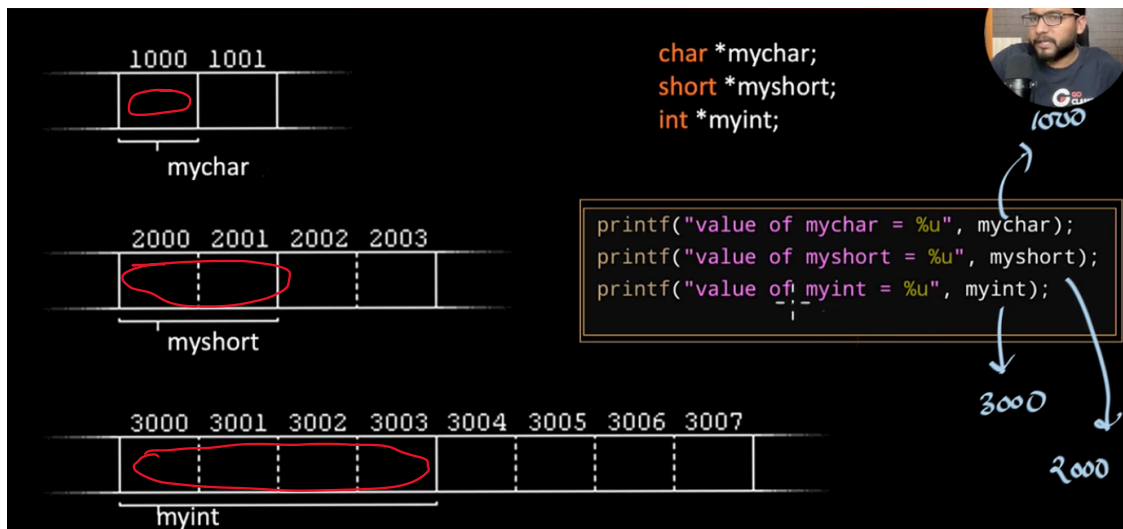


$*p1 = *p2$

causes p2 to start pointing to what ever p2 points to.

Only address of first box is stored

Only address of first byte is stored



Only first byte is taken.