C Programming

and

Debugging

CS 35L

Spring 2020 – Section 7

Assignment 10

- Form teams of 2
- Choose a recent CS story
- Presentation + Review
- Use the Google Sheet to add your names and check what stories other teams have chosen.
- No teammate? Add name to random list column and you will be randomly paired.

Pointers review

- Variables that store memory addresses

- **Declaration:** <variable_type> *<name>;
  - int *ptr; //declare ptr as a pointer to int
  - int var = 77; // define an int variable
  - ptr = &var; // let ptr point to the variable var

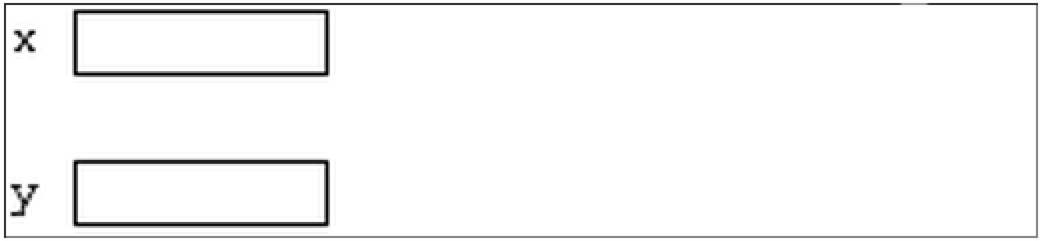
(De)Referencing

- **Referencing**: get the address of a variable
- **Dereferencing**: getting the value that the pointer is currently pointing to

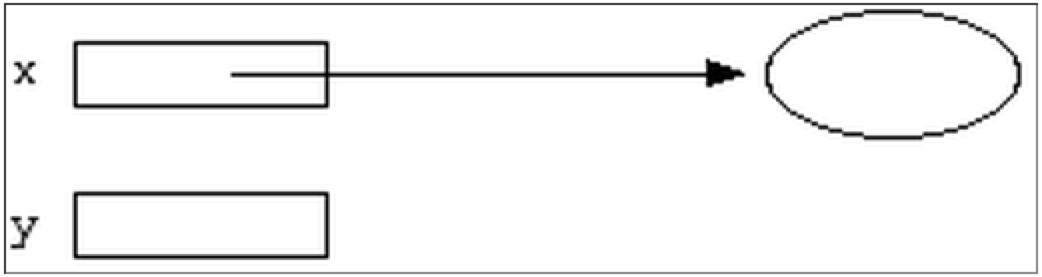
- Example:
  - double x, *ptr;
  - ptr = **&**x; //referencing: let ptr point to x
  - *ptr = 7.8; //dereferencing: assign 7.8 to x

Pointer Example

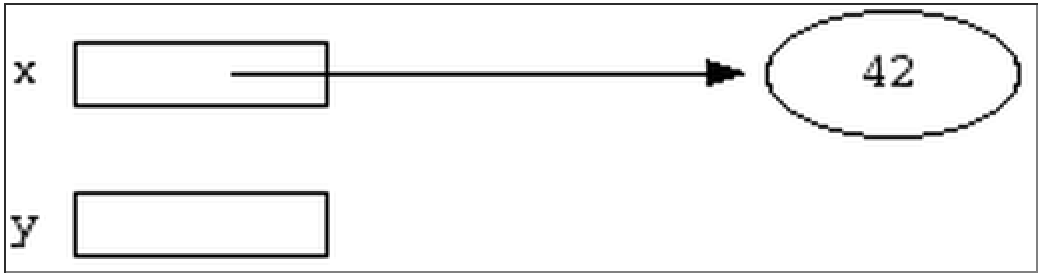
```
int *x;

int *y;
```



```
int var;   x = &var;
```

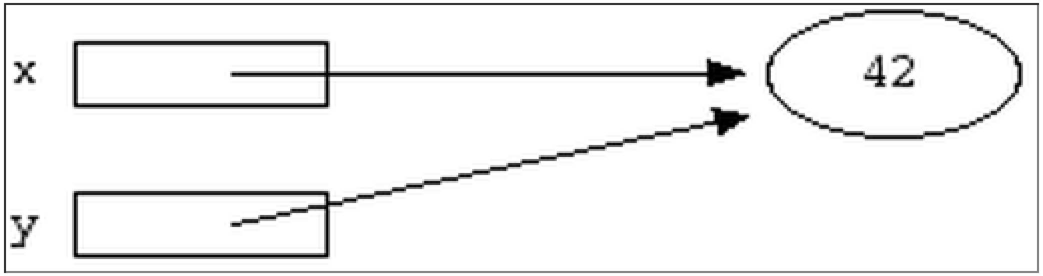


```
*x = 42;
```

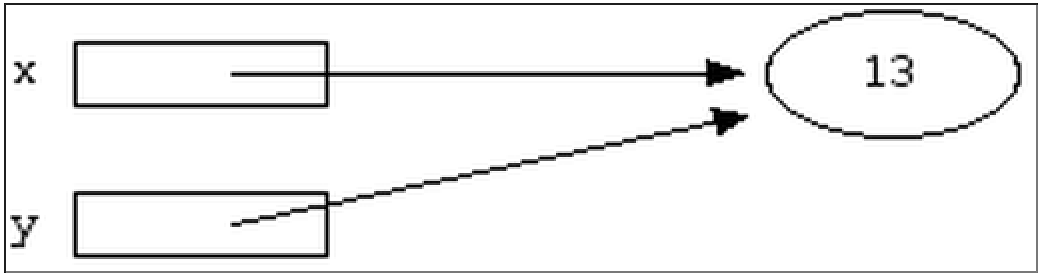


Pointer Example

```
y = x;
```

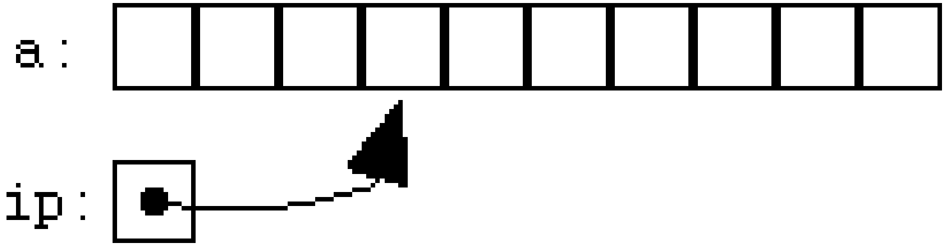


```
*x = 13;     or
*y = 13;
```

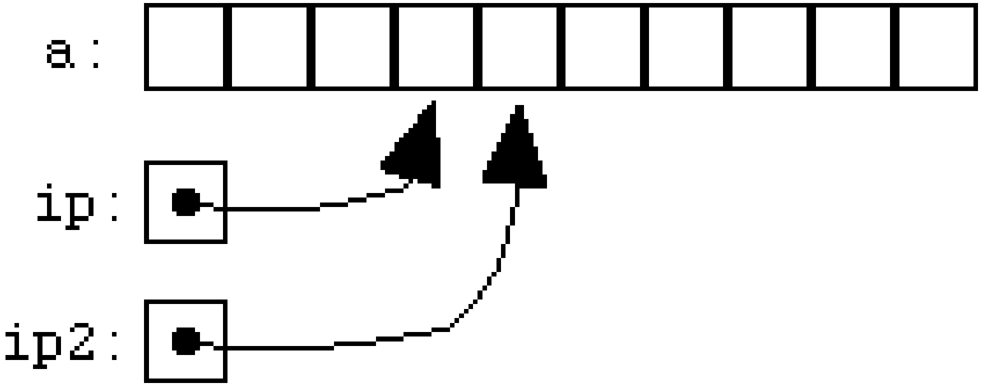


Pointer Arithmetic

```
int *ip;
int a[10];
ip = &a[3];
```



```
int *ip2;
ip2 = ip + 1;
*ip2 = 4;

*(ip + 3) = 7;
ip += 1;
ip++;
```



Pointers to Functions

- Also known as: **function pointers** or **functors**
- Goal: write a sorting function
  - Has to work for ascending and descending sorting order + other
- How?
  - Write multiple functions
  - Provide a flag as an argument to the function
  - Polymorphism and virtual functions
  - Use function pointers!!

Pointers to Functions

- Declaration

```
double (*func_ptr) (double, double);
func_ptr = &pow;
func_ptr = pow;
```

- Usage:

```
double result = (*func_ptr)( 1.5, 2.0 );
        double result = func_ptr( 1.5, 2.0 );
```

qsort Example

Q-sort (Standard C Library) function prototype:

```
void qsort (void* base, size_t num, size_t size,
            int (*comparator)(const void*,const void*));


//void* is a pointer that points to any data type!
//void* cannot be dereferenced - must be type casted first
//No pointer arithmetic with void*
```

qsort Example

```
int compare (const void * a, const void * b) {
      return ( *(int*)a - *(int*)b );
}

//(int*) type casts ‘a’ to an int pointer
//*(int*) dereferences the pointer (access value)

int main () {
      int values[] = { 40, 10, 100, 90, 20, 25 };
      qsort (values, 6, sizeof(int), compare);
      int n;
      for (n = 0; n < 6; n++)
            printf ("%d ",values[n]);
      return 0;
}
```

Dynamic Memory

- Memory that is allocated at runtime
- Allocated on the **heap**

void *malloc (size_t size);

- Allocates *size* bytes and returns a pointer to the allocated memory

void *realloc (void *ptr, size_t size);

- Changes the size of the memory block pointed to by *ptr* to *size* bytes

void free (void *ptr);

- Frees the block of memory pointed to by *ptr*

Valgrind

- Powerful dynamic analysis tool
- Useful to detect memory leaks

Example:

```
$ valgrind --leak-check=full
        ./sfrob < foo.txt
```

```
88 (...) bytes in 1 blocks are definitely lost ...
   at 0x........: malloc (vg_replace_malloc.c:...)
   by 0x........: mk (leak-tree.c:11)
   by 0x........: main (leak-tree.c:25)
```

Lab Assignment

- Install an old version of coreutils
- **ls** has a bug (again!)
- **ls** –lt
- -rw-r--r-- 1 eggert csfac 0 1918-11-11 11:00:00.000000000 +0000 wwi-armistice-cs35L
- -rw-r--r-- 1 eggert csfac 0 2018-10-29 16:43:16.805404419 +0000 now1
- -rw-r--r-- 1 eggert csfac 0 2018-10-29 16:43:15.801376773 +0000 now

- Build old coreutils (will fail, report why)
- Apply patch to coreutils (why did it work?)
- Reproduce error and debug (hint: what function in ls.c compares timestamps?)
- Fix the error and create a patch (diff –u)
- Reproduce the error on SEASnet (why is it bugging yet again?)