Lecture 15: Dynamic Memory Allocation (malloc/heap)

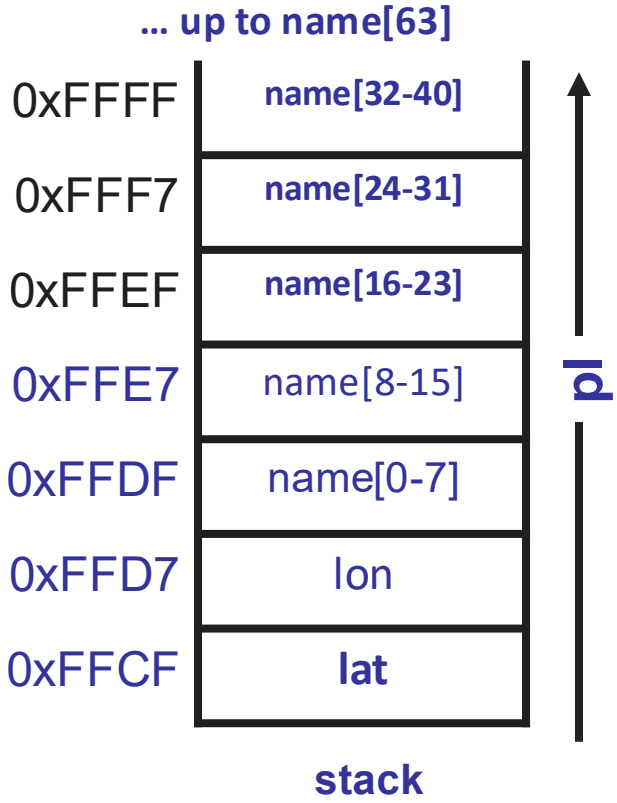
CSE 29: Systems Programming and Software Tools

Aaron Schulman (Shalev)

struct memory layout

```
struct place {
    long int lat; // Latitude
    long int lon; // Longitude
    char name[64]; // Name
};

int main() {
 struct place pl;
 pl.lat = 10315;
 pl.lon = 11561;
 return 0;
}
```



Dynamic memory allocation

We have been statically initializing the size of an array at *compile time*:

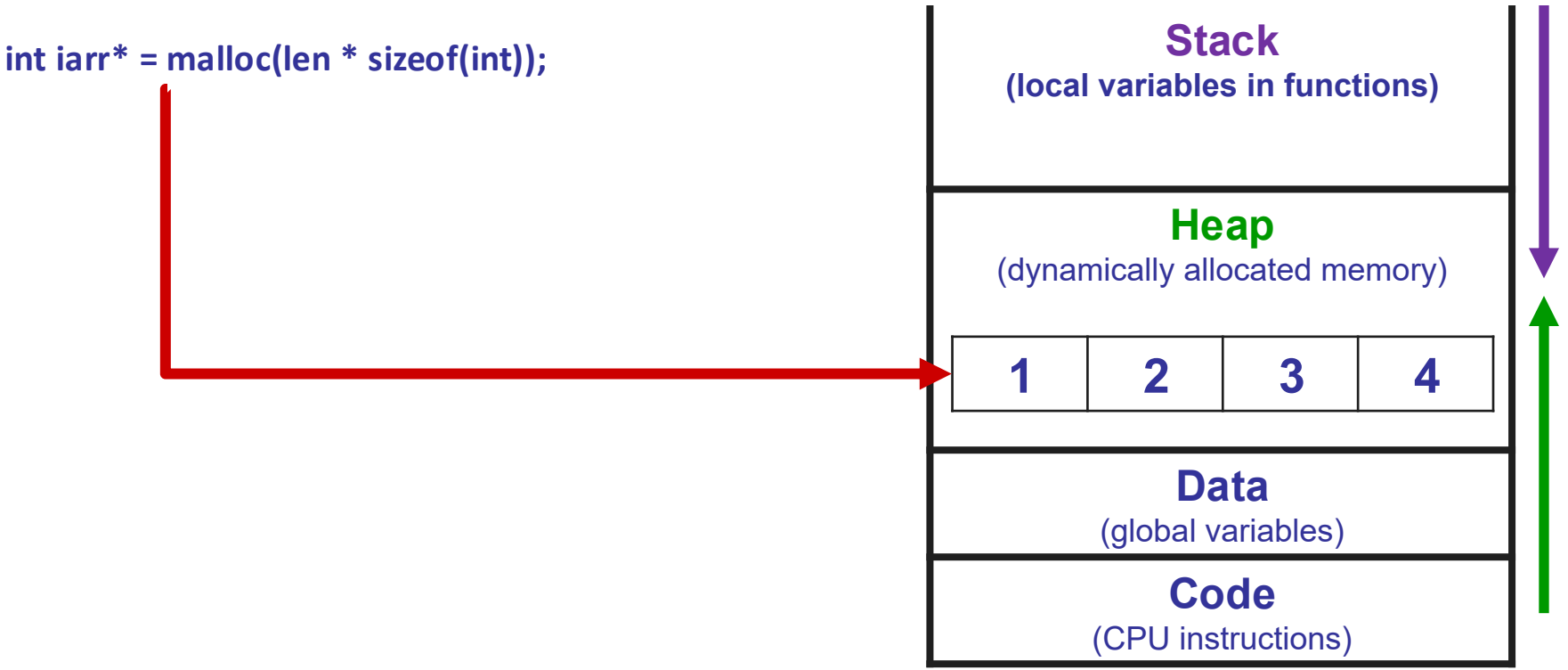
```
int arr[4]; // The length 4 is defined at compile time
```

What if we want to create an array where the size is defined at *runtime*?

```
int len = 10; // Can be changed at runtime
int *iarr = NULL;
iarr = malloc(len * sizeof(int));

iarr[8] = 5;  // or *(iarr + 8) = 5
```

Where are malloc() allocations in memory?



Dynamic memory allocation

- When you know what the size of arrays, use the stack:
  - The compiler automatically makes room for them on the stack
- When you don't know the size of arrays at compile time, you need to allocate them from another memory region
  - The Heap: Memory region in a program for dynamically sized arrays
    - » The heap will need to be managed (you will implement this in this class!)
    - » You can change the amount of memory used to store an array during the execution of your program