Lecture 33: More pointers

Michael J. Bannister and Kim Bruce
Pomona College, Fall 2015

Pointer Example

(white board)

Assignments this week

Assignment 11: Animal game

- Due next 11/10
- Working in pairs suggested
- Start early!

Lab 11: Linked lists

- Convert sinlge linked list to double
- Good practice for the assignment
- Learn to use valgrind

Dynamic memory management in C

Allocating memory:

- Use `void* malloc( size_t size );`
- NULL returned if allocation fails (need to check this!)
- Look up, but dont use `calloc` and `realloc`

Deallocating memory

- Use `void free(void* ptr)`
- Cannot fail
- Only deallocate memory allocated by `mallo`
- Don't double deallocate memory
- Don't read or write to deallocated memory

Pointer arithmetic

Suppose `T* ptr = &A[i]` where A has type T[], i.e., ptr points to the ith entry of A.

- `ptr+n = n+ptr` = a pointer the array element with index i+n
- `ptr-n` = a pointer to array element with index i-n

Now suppose `ptr1 = &A[i]` and `ptr2 = &A[j]`

- `ptr1-ptr2` = the value j - i

For this to work all pointers must be valid!

The const type qualifier

The `const` keyword is used to make a variable read only, i.e., to define constants.

Exmaple:

```
const int x = 1;
x = 2; // ERROR
int y = 2;
const int* p = &y;
*p = 3; // ERROR
p = &x; // OK
```

```
// continued
int * const q = &y;
*q = 4; // OK
q = p; // ERROR
```

Note: `"const int"` and `"int const"` mean the same thing.

Type unsafe generics

- Void pointers, `void*`, can point to anything!
- So they can be used to implement type unsafe generic data structures and algorithms
- This is very dangerous, as types are not checked at all.
- Need to use functions pointer to pass functions as arguments to other functions

Example: quick sort from `stdlib.h`

```
void qsort( void *ptr, size_t count, size_t size,
        int (*comp)(const void *, const void *));
```

Explanation on board.