

Systems Programming (Spring 2017)

Assignment 1: Memory Allocation++

Contributors: Enkai Ji & Joseph Moussa

Custom implementation of malloc() and free() library calls for dynamic memory allocation that detect common programming and usage errors.

Build

```
$ make
```

Run the test

```
$ ./memgrind
```

Usage of the custom malloc() and free()

```
#include "mymalloc.h"
void* p;
p = malloc(1);
free(p);
```

Testing

Below is result of four tests. And each one is executed 100 times and the time cost is recorded/

Test1:

1000 separate malloc()s of 1 byte, then free() the 1000 1 byte pointers one by one

Costs 0.530000 seconds

Test2:

first malloc() 1 byte and immediately free it - do this 1000 times

Costs 0.266000 seconds

Test3:

Randomly choose between a 1 byte malloc() or free()ing a 1 byte pointer - do this 1000 times

Costs 0.046000 seconds

Test4:

Randomly choose between a randomly-sized malloc() or free()ing a pointer - do this many times

Costs 0.047000 seconds

Error handling

We expect

- Free()ing addresses that are not pointers
- Redundant free()ing of the same pointer