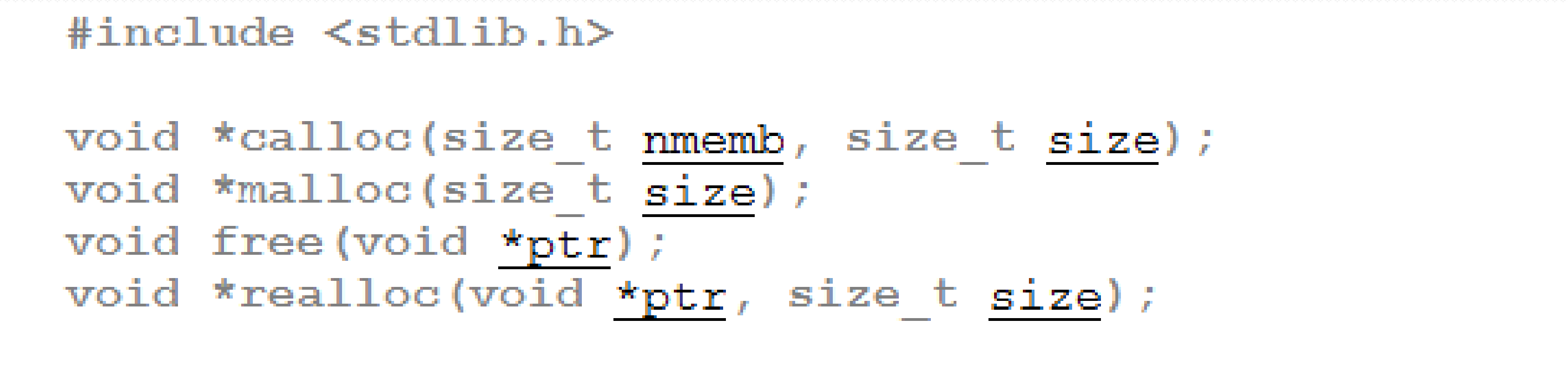
Dynamic Memory Allocation & Memory Leaks

15-123

Effective Programming in C and Unix

malloc, calloc, realloc & free



```
#include <stdlib.h>

void *calloc(size_t nmemb, size_t size);
void *malloc(size_t size);
void free(void *ptr);
void *realloc(void *ptr, size_t size);
```

Things to note about malloc group of functions

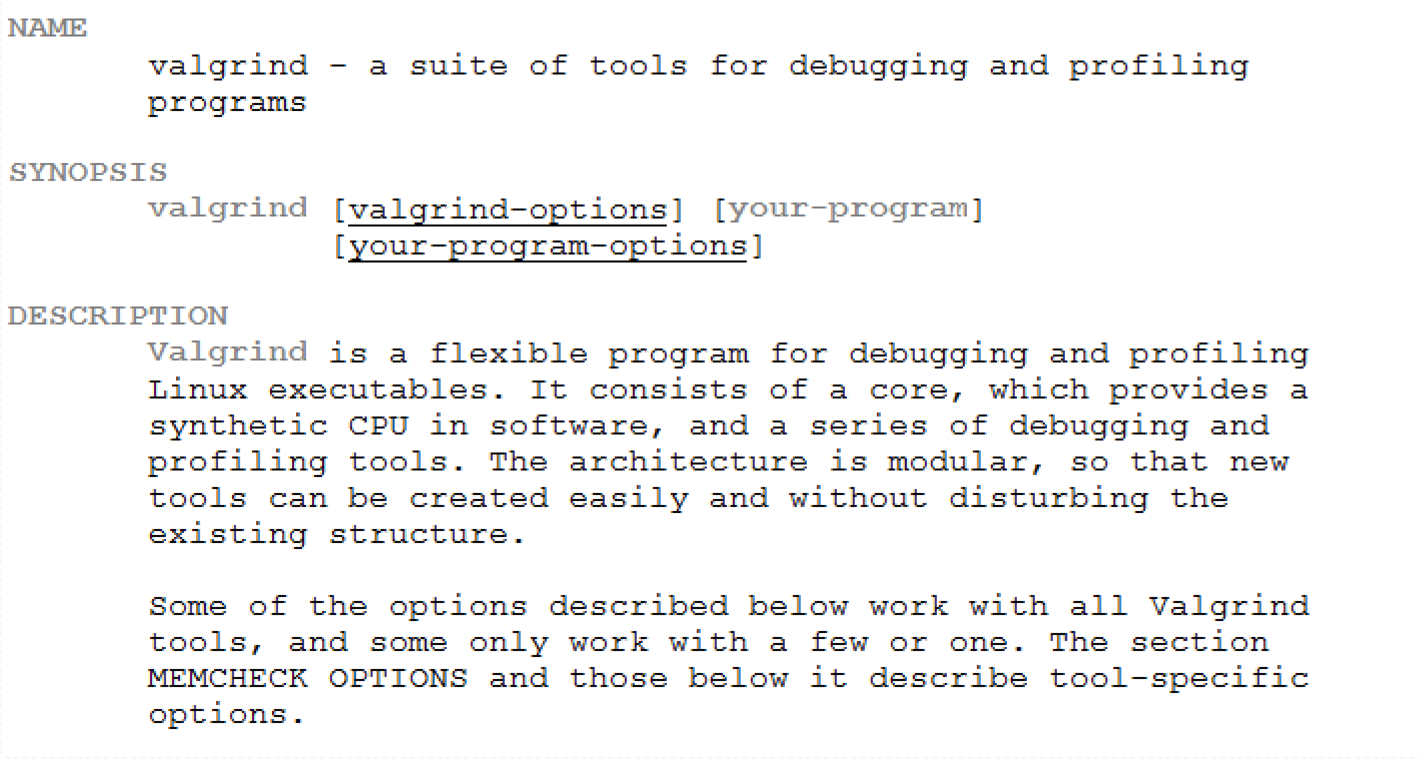
- malloc returns a void* , that is pointer with no designation of its type
  - Can be assigned to any other type
  - int* ptr = malloc(12);
- calloc is like malloc, except that block of memory returned is initialized
- realloc is a convenient way to resize an array, but it does not guarantee that the starting address of the block is invariant
- free will deallocate the block of memory using any pointer to the block.

Memory Leaks

- Each program during its execution are allocated memory from stack and heap
  - stack refers to memory locations used by compiler during its execution of the program
  - heap refers to the memory locations that are being used by compiler to allocate dynamic memory allocation requests from the program (eg: malloc)
- Memory leaks refers to programs leaving unused memory in the heap
- All heap memory must be exclusively freed
- Example of memory leak:
  - int* ptr = malloc(4);
  - int x; ptr = &x;

Detecting memory leaks

- A useful tool for detecting memory leaks is valgrind
    - www.valgrind.org



```
NAME
       valgrind - a suite of tools for debugging and profiling
       programs

SYNOPSIS
       valgrind [valgrind-options] [your-program]
                [your-program-options]

DESCRIPTION
       Valgrind is a flexible program for debugging and profiling
       Linux executables. It consists of a core, which provides a
       synthetic CPU in software, and a series of debugging and
       profiling tools. The architecture is modular, so that new
       tools can be created easily and without disturbing the
       existing structure.

       Some of the options described below work with all Valgrind
       tools, and some only work with a few or one. The section
       MEMCHECK OPTIONS and those below it describe tool-specific
       options.
```

> valgrind --tool=memcheck --leak-check=full ./a.out

Coding Examples