MONTANA
STATE UNIVERSITY

Programming with C I

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

2024.04.24

What is Valgrind?

- **Valgrind is a flexible program for debugging and profiling Linux executables. Most important usage of valgrind is when you try to find memory leaks.**

Download valgrind

```
$ sudo apt-get install valgrind
```

How to use a Valgrind

- **You can choose what tool you want to use with Valgrind. I explain only memcheck and massif which are mostly used and important.**

--tool=<toolname> [default: memcheck]

Run the Valgrind tool called toolname,

e.g. **memcheck**, cachegrind, callgrind, helgrind, drd, **massif**, lackey, none,
exp-sgcheck,exp-bbv, exp-dhat, etc.

memcheck

- **Memcheck is a memory error detector. It can detect the following problems that are common in C and C++ programs.**

- **Accessing memory you shouldn’t:** e.g. overrunning and underrunning heap blocks, overrunning the top of the stack, and accessing memory after it has been freed.
- **Using undefined values:** i.e. values that have not been initialised, or that have been derived from other undefined values.
- **Incorrect freeing of heap memory: s**uch as double-freeing heap blocks, or mismatched use of malloc/new/new[] versus free/delete/delete[]

memcheck

- **Memcheck is a memory error detector. It can detect the following problems that are common in C and C++ programs.**

- **Overlapping src and dst pointers in memcpy and related functions.**
- **Passing a fishy (presumably negative) value to the size parameter of a memory allocation function.**
- **Memory leaks.**

massif

- **Massif is a heap profiler. It measures how much heap memory your program uses. This includes both the useful space, and the extra bytes allocated for book-keeping and alignment purposes. It can also measure the size of your program’s stack(s), although it does not do so by default.**
- **In summary, it can give you information about:**
  - Heap blocks
  - Heap administration blocks
  - Stack sizes.

Run for a real use case

Memory leak example

```
#include <stdio.h>
int main()
{
    while(true) {
        int* a = new int[100];
        sleep(1);
    }
}
```

Run for a real use case

Run valigrind to check memory leak

```
$ valgrind --tool=memcheck --leak-check=yes --leak-check=full --show-leak-kinds=all -v ./a.out
```

check heap and stack with massif

```
$ valgrind --tool=massif --stacks=yes --log-file=valgrind.log ./a.out
```

- **running valgrind with above will create two files, one massif.out.xxx file and the other valgrind.log file.**

THE END

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

2024.04.24