Report Case 3

Pamela Salazar

May 2023

1 Introduction

Valgrind is an instrumentation framework for building dynamic analysis tools. It comes with a set of tools each of which performs some kind of debugging, profiling, or similar task that helps you improve your programs. Valgrind's architecture is modular, so new tools can be created easily and without disturbing the existing structure. Memcheck is a memory error detector in Valgrind. It helps you make your programs, particularly those written in C and C++, more correct. [1].

For example the C code of the case 3 has 9 allocations, once for anytime the for loop is running. The size of the int type is 4 bytes, 4 multiplied by 100 is equal to 400 bytes. The 400 bytes allocates 9 times, means 3600 bytes allocated.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv){
  int i;
  int *a;

  for (i=0; i < 10; i++){
    a = malloc(sizeof(int) * 100);
  }
  free(a);
  return 0;
}
```

Valgrind reports the 3600 bytes use when the program exits, the 9 memory allocations and the n1 free in the next way.

```
==18831== Memcheck, a memory error detector
==18831== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==18831== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==18831== Command: ./case3
==18831==
==18831==
```

```
==18831== HEAP SUMMARY:
==18831==     in use at exit: 3,600 bytes in 9 blocks
==18831==   total heap usage: 10 allocs, 1 frees, 4,000 bytes allocated
==18831==
==18831== LEAK SUMMARY:
==18831==    definitely lost: 3,600 bytes in 9 blocks
==18831==    indirectly lost: 0 bytes in 0 blocks
==18831==      possibly lost: 0 bytes in 0 blocks
==18831==    still reachable: 0 bytes in 0 blocks
==18831==         suppressed: 0 bytes in 0 blocks
==18831== Rerun with --leak-check=full to see details of leaked memory
==18831==
==18831== For lists of detected and suppressed errors, rerun with: -s
==18831== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

References

[1] Valgrind Manual User. `https://valgrind.org/docs/manual/manual.html`, 24^{th} may 2023.