

# Programming Assignment 3 Part A

---

Binghan Geng

A20482350

[bgeng1@hawk.iit.edu](mailto:bgeng1@hawk.iit.edu)

Yu Li

A20496405

[yli385@hawk.iit.edu](mailto:yli385@hawk.iit.edu)

## Introduction

This part is to learn how to use the debugging tools gdb and valgrind to find memory leaks and other insidious memory problems.

## Build & Run Instructions

execute by gdb:

```
cs450@ubuntu:~/PA3_part1$ gcc mem_leak.c -g -o mem_leak ← 1
cs450@ubuntu:~/PA3_part1$ gdb ./mem_leak ← 2
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./mem_leak...done.
(gdb) □
```

execute by valgrind:

Instruction 1: valgrind --leak-check=yes ./mem\_leak

```
cs450@ubuntu:~/PA3_part1$ valgrind --leak-check=yes ./mem_leak
==8357== Memcheck, a memory error detector
==8357== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==8357== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==8357== Command: ./mem_leak
==8357==
==8357==
==8357== HEAP SUMMARY:
==8357==   in use at exit: 1,024 bytes in 1 blocks
==8357==   total heap usage: 2 allocs, 1 frees, 1,049,600 bytes allocated
==8357==
==8357== 1,024 bytes in 1 blocks are definitely lost in loss record 1 of 1
==8357==   at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==8357==   by 0x1086A1: test_memory_leak (mem_leak.c:5)
==8357==   by 0x1086EC: main (mem_leak.c:15)
==8357==
==8357== LEAK SUMMARY:
==8357==   definitely lost: 1,024 bytes in 1 blocks
==8357==   indirectly lost: 0 bytes in 0 blocks
==8357==   possibly lost: 0 bytes in 0 blocks
==8357==   still reachable: 0 bytes in 0 blocks
==8357==   suppressed: 0 bytes in 0 blocks
==8357==
==8357== For counts of detected and suppressed errors, rerun with: -v
==8357== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

Instruction 2: valgrind --xtree-leak=yes ./mem\_leak

```
cs450@ubuntu:~/PA3_part1$ valgrind --xtree-leak=yes ./mem_leak
==5474== Memcheck, a memory error detector
==5474== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==5474== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==5474== Command: ./mem_leak
==5474==
==5474==
==5474== HEAP SUMMARY:
==5474==   in use at exit: 1,024 bytes in 1 blocks
==5474==   total heap usage: 2 allocs, 1 frees, 1,049,600 bytes allocated
==5474==
==5474== xtree leak report: /home/cs450/PA3_part1/xtleak.kcg.5474
==5474== LEAK SUMMARY:
==5474==   definitely lost: 1,024 bytes in 1 blocks
==5474==   indirectly lost: 0 bytes in 0 blocks
==5474==   possibly lost: 0 bytes in 0 blocks
==5474==   still reachable: 0 bytes in 0 blocks
==5474==   suppressed: 0 bytes in 0 blocks
==5474==
==5474== For counts of detected and suppressed errors, rerun with: -v
==5474== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

Instruction 3: valgrind --show-reachable=yes ./mem\_leak

```
cs450@ubuntu:~/PA3_part1$ valgrind --show-reachable=yes ./mem_leak
==5522== Memcheck, a memory error detector
==5522== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==5522== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==5522== Command: ./mem_leak
==5522==
==5522==
==5522== HEAP SUMMARY:
==5522==   in use at exit: 1,024 bytes in 1 blocks
==5522==   total heap usage: 2 allocs, 1 frees, 1,049,600 bytes allocated
==5522==
==5522== LEAK SUMMARY:
==5522==    definitely lost: 1,024 bytes in 1 blocks
==5522==    indirectly lost: 0 bytes in 0 blocks
==5522==    possibly lost: 0 bytes in 0 blocks
==5522==    still reachable: 0 bytes in 0 blocks
==5522==    suppressed: 0 bytes in 0 blocks
==5522== Rerun with --leak-check=full to see details of leaked memory
==5522==
==5522== For counts of detected and suppressed errors, rerun with: -v
==5522== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```