

HW3 Report

Name: Xiaowei Tan

ID: 69203272

Screenshots for script execution

```
2. xiaowei@andromeda-27:~/253p/hw/hw3 (ssh)
$ script hw3.script
Script started, file is hw3.script
xiaowei@andromeda-27 23:39:29 ~/253p/hw/hw3
$ make clean
rm -rf countChar countChar.o main.o
xiaowei@andromeda-27 23:39:39 ~/253p/hw/hw3
$ make
g++ -ggdb -Wall -o countChar.o -c countChar.cpp
g++ -ggdb -Wall -o main.o -c main.cpp
g++ -ggdb -o countChar countChar.o main.o
-----OK-----
xiaowei@andromeda-27 23:39:41 ~/253p/hw/hw3
$ gdb ./countChar
GNU gdb (GDB) 7.12
Copyright (C) 2016 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-pc-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 ./countChar...done.
(gdb) br main
Breakpoint 1 at 0x40074c: file main.cpp, line 4.
(gdb) run < sample
Starting program: /home/xiaowei/253p/hw/hw3/countChar < sample
warning: File "/pkg/gcc/5.4.0/lib64/libstdc++.so.6.0.21-gdb.py" auto-loading has been declined by your `auto-load safe-path' set to "$debugdir:$datadir/auto-load:/home/xiaowei/cs238p/xv6-public/.gdbinit".
To enable execution of this file add
    add-auto-load-safe-path /pkg/gcc/5.4.0/lib64/libstdc++.so.6.0.21-gdb.py
line to your configuration file "/home/xiaowei/.gdbinit".
To completely disable this security protection add
    set auto-load safe-path /
line to your configuration file "/home/xiaowei/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual. E.g., run from the shell:
    info "(gdb)Auto-loading safe path"
```

```
2. xiaowei@andromeda-27:~/253p/hw/hw3 (ssh)
Breakpoint 1, main () at main.cpp:4
4      doCount();
(gdb) s
doCount () at countChar.cpp:25
25      int printable[95] = {0};
(gdb) n
26      int space = 0;
(gdb)
27      int other = 0;
(gdb)
28      count(printable, space, other);
(gdb) s
count (printable=0x7fffffff800, space=@0x7fffffff7fc: 0, other=@0x7fffffff7f8: 0) at countChar.cpp:6
6      while((c = getchar()) != EOF){
(gdb) n
7          if(c >= ' ' && c <= '~')
(gdb)
8              printable[c-' ' ]++;
(gdb)
9          if(c == ' ' || c == '\t' || c == '\n')
(gdb) print c
$1 = 79 '0'
(gdb) backtrace
#0 count (printable=0x7fffffff800, space=@0x7fffffff7fc: 0, other=@0x7fffffff7f8: 0) at countChar.cpp:9
#1 0x00000000400725 in doCount () at countChar.cpp:28
#2 0x00000000400751 in main () at main.cpp:4
(gdb)
#0 count (printable=0x7fffffff800, space=@0x7fffffff7fc: 0, other=@0x7fffffff7f8: 0) at countChar.cpp:9
#1 0x00000000400725 in doCount () at countChar.cpp:28
#2 0x00000000400751 in main () at main.cpp:4
(gdb) n
11      if(c < ' ' || c > '~')
(gdb)
6      while((c = getchar()) != EOF){
(gdb) continue
Continuing.
^_: 91
^": 10
^,: 28
^_: 1
^,: 7
^": 1
```

```

'?: 1
'C': 2
'E': 1
'H': 1
'I': 2
'M': 1
'O': 2
'T': 4
'W': 1
'a': 27
'b': 6
'c': 5
'd': 19
'e': 69
'f': 8
'g': 14
'h': 22
'i': 36
'j': 2
'k': 4
'l': 21
'm': 6
'n': 35
'o': 37
'p': 11
'r': 26
's': 32
't': 35
'u': 11
'v': 6
'w': 6
'y': 13
whitespace characters: 110
other characters: 19
[Inferior 1 (process 27098) exited normally]
(gdb) quit
xiaoweit@andromeda-27 23:41:57 ~/253p/hw/hw3
$ make clean
rm -rf countChar countChar.o main.o
xiaoweit@andromeda-27 23:42:00 ~/253p/hw/hw3
$ cat Makefile
CXXFLAGS=-std=c++11 -Wpedantic -Wall -Wextra -Werror -Weffc++
CFLAGS=-std=c99 -pedantic -Wall -Wextra -Werror -ansi -Wwrite-strings -Weffc++

.SUFFIXES:.cpp .o

CC=g++

SRCS=countChar.cpp \
    main.cpp

OBJS=$(SRCS:.cpp=.o)
EXEC=countChar

start: $(OBJS)
    $(CC) -ggdb -o $(EXEC) $(OBJS)
    @echo "-----OK-----"

.cpp.o:
    $(CC) -ggdb -Wall -o $@ -c $<

clean:
    rm -rf $(EXEC) $(OBJS)
xiaoweit@andromeda-27 23:42:11 ~/253p/hw/hw3
$ exit
exit
Script done, file is hw3.script
xiaoweit@andromeda-27 23:42:17 ~/253p/hw/hw3
$

```

Screenshots for 3 problems

1. parseURL

```
2. xiaowei@andromeda-27:~/253p/hw/hw3 (ssh)
$ ./parseURL
https://apis.google.com/js/client.js
scheme:https
domain:apis.google.com
port:
path:js/client.js
queryString:
fragmentId:

http://www.facebook.com/pages/Learn-the-Net/330002341216
scheme:http
domain:www.facebook.com
port:
path:pages/Learn-the-Net/330002341216
queryString:
fragmentId:

ftp://rtfm.mit.edu/pub/
scheme:ftp
domain:rtfm.mit.edu
port:
path:pub/
queryString:
fragmentId:

https://www.google.com/search?q=ebay
scheme:https
domain:www.google.com
port:
path:search
queryString:q=ebay
fragmentId:

www.learnthenet.com/web-at-a-glance/url-examples/
scheme:
domain:www.learnthenet.com
port:
path:web-at-a-glance/url-examples/
queryString:
fragmentId:
```

```
http://my.ebay.com:80/ws/eBayISAPI.dll?MyEbayBeta&MyEbay=&gbh=1&guest=1#myfragment
scheme:http
domain:my.ebay.com
port:80
path:ws/eBayISAPI.dll
queryString:MyEbayBeta&MyEbay=&gbh=1&guest=1
fragmentId:myfragment
```

```
2. xiaowei@andromeda-27:~/253p/hw/hw3 (ssh)
$ valgrind ./parseURL
==28778== Memcheck, a memory error detector
==28778== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==28778== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==28778== Command: ./parseURL
==28778==
http://my.ebay.com:80/ws/eBayISAPI.dll?MyEbayBeta&MyEbay=&gbh=1&guest=1#myfragment
scheme:http
domain:my.ebay.com
port:80
path:ws/eBayISAPI.dll
queryString:MyEbayBeta&MyEbay=&gbh=1&guest=1
fragmentId:myfragment

^C==28778==
==28778== Process terminating with default action of signal 2 (SIGINT)
==28778== at 0x5788F70: __read_nocancel (in /usr/lib64/libc-2.17.so)
==28778== by 0x5745B13: _IO_file_underflow@GLIBC_2.2.5 (in /usr/lib64/libc-2.17.so)
==28778== by 0x5746CE1: _IO_default_uflow (in /usr/lib64/libc-2.17.so)
==28778== by 0x574154D: getc (in /usr/lib64/libc-2.17.so)
==28778== by 0x4F1B13C: syncgetc (stdio_sync_filebuf.h:225)
==28778== by 0x4F1B13C: __gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char> >::underflow() (stdio_sync_filebuf.h:133)
==28778== by 0x4EDB4B9: sgetc (streambuf:344)
==28778== by 0x4EDB4B9: std::basic_istream<char, std::char_traits<char> >& std::getline<char, std::char_traits<char>, std::allocator<char> >(std::basic_istream<char, std::char_traits<char> >&, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >&, char) (istream-string.cc:145)
==28778== by 0x401880: main (in /home/xiaowei/253p/hw/hw3/parseURL)
==28778==
==28778== HEAP SUMMARY:
==28778== in use at exit: 72,950 bytes in 5 blocks
==28778== total heap usage: 125 allocs, 120 frees, 81,184 bytes allocated
==28778==
==28778== LEAK SUMMARY:
==28778== definitely lost: 0 bytes in 0 blocks
==28778== indirectly lost: 0 bytes in 0 blocks
==28778== possibly lost: 0 bytes in 0 blocks
==28778== still reachable: 72,950 bytes in 5 blocks
==28778== suppressed: 0 bytes in 0 blocks
==28778== Rerun with --leak-check=full to see details of leaked memory
==28778==
==28778== For counts of detected and suppressed errors, rerun with: -v
==28778== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

xiaowei@andromeda-27 23:54:49 ~/253p/hw/hw3
```

2. countChar

```
2. xiaowei@andromeda-27:~/253p/hw/hw3 (ssh)
$ ./countChar < sample
' ': 91
'\"': 10
',': 28
'-': 1
'.': 7
':': 1
'?': 1
'C': 2
'E': 1
'H': 1
'I': 2
'M': 1
'O': 2
'T': 4
'W': 1
'a': 27
'b': 6
'c': 5
'd': 19
'e': 69
'f': 8
'g': 14
'h': 22
'i': 36
'j': 2
'k': 4
'l': 21
'm': 6
'n': 35
'o': 37
'p': 11
'r': 26
's': 32
't': 35
'u': 11
'v': 6
'w': 6
'y': 13
whitespace characters: 110
other characters: 19
xiaowei@andromeda-27 23:51:10 ~/253p/hw/hw3
$
```

```
2. xiaowei@andromeda-27:~/253p/hw/hw3 (ssh)
$ valgrind countChar < sample
==28854== Memcheck, a memory error detector
==28854== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==28854== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==28854== Command: countChar
==28854==
' ': 91
'\"': 10
',': 28
'-': 1
'.': 7
':': 1
'?': 1
'C': 2
'E': 1
'H': 1
'I': 2
'M': 1
'O': 2
'T': 4
'W': 1
'a': 27
'b': 6
'c': 5
'd': 19
'e': 69
'f': 8
'g': 14
'h': 22
'i': 36
'j': 2
'k': 4
'l': 21
'm': 6
'n': 35
'o': 37
'p': 11
'r': 26
's': 32
't': 35
'u': 11
'v': 6
'w': 6
```

```
'y': 13
whitespace characters: 110
other characters: 19
==28854==
==28854== HEAP SUMMARY:
==28854==   in use at exit: 72,704 bytes in 1 blocks
==28854==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==28854==
==28854== LEAK SUMMARY:
==28854==   definitely lost: 0 bytes in 0 blocks
==28854==   indirectly lost: 0 bytes in 0 blocks
==28854==   possibly lost: 0 bytes in 0 blocks
==28854==   still reachable: 72,704 bytes in 1 blocks
==28854==   suppressed: 0 bytes in 0 blocks
==28854== Rerun with --leak-check=full to see details of leaked memory
==28854==
==28854== For counts of detected and suppressed errors, rerun with: -v
==28854== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaowei@andromeda-27 23:55:26 ~/253p/hw/hw3
$
```

3. convertRoman

```
2. xiaowei@andromeda-27:~/253p/hw/hw3 (ssh)
$ ./convert
II
II => 2
IV
IV => 4
VIII
VIII => 8
XVI
XVI => 16
XXXII
XXXII => 32
LXIV
LXIV => 64
CXXVIII
CXXVIII => 128
CCLVI
CCLVI => 256
DXII
DXII => 512
MXXIV
MXXIV => 1024
MMXLVIII
MMXLVIII => 2048
MMMMXCVI
MMMMXCVI => 4096
MMMMMMXCII
MMMMMMXCII => 8192
$ valgrind ./convert
==28938== Memcheck, a memory error detector
==28938== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==28938== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==28938== Command: ./convert
==28938==
MMMMMMXCII
MMMMMMXCII => 8192
^C==28938==
==28938== Process terminating with default action of signal 2 (SIGINT)
==28938== at 0x5788F70: __read_nocancel (in /usr/lib64/libc-2.17.so)
==28938== by 0x5745B13: _IO_file_underflow@@GLIBC_2.2.5 (in /usr/lib64/libc-2.17.so)
==28938== by 0x5746CE1: _IO_default_uflow (in /usr/lib64/libc-2.17.so)
==28938== by 0x574154D: getc (in /usr/lib64/libc-2.17.so)
==28938== by 0x4F1B13C: syncgetc (stdio_sync_filebuf.h:225)
==28938== by 0x4F1B13C: __gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char> >::underflow() (stdio_sync_filebuf.h:133)
==28938== by 0xEDB4B9: sgetc (streambuf:344)
==28938== by 0xEDB4B9: std::basic_istream<char, std::char_traits<char> >& std::getline<char, std::char_traits<char>, std::allocator<char> >(std::basic_istream<char, std::char_traits<char> >&, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >&, char) (istream-string.cc:145)
==28938== by 0x400FFA: main (in /home/xiaowei/253p/hw/hw3/convert)
==28938==
==28938== HEAP SUMMARY:
==28938==   in use at exit: 72,704 bytes in 1 blocks
==28938== total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==28938==
==28938== LEAK SUMMARY:
==28938==   definitely lost: 0 bytes in 0 blocks
==28938==   indirectly lost: 0 bytes in 0 blocks
==28938==   possibly lost: 0 bytes in 0 blocks
==28938==   still reachable: 72,704 bytes in 1 blocks
==28938==   suppressed: 0 bytes in 0 blocks
==28938== Rerun with --leak-check=full to see details of leaked memory
==28938==
==28938== For counts of detected and suppressed errors, rerun with: -v
==28938== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaowei@andromeda-27 23:56:37 ~/253p/hw/hw3
$
```