Exam 1 Report

ECPE 170 – Computer Systems and Networks – Spring 2016

Name: Drew Overgaard

Exam 1

Question a:
Answer:
To download the file use the following command:

```
wget http://www1.pacific.edu/~jshafer3/ecpe170/exam1.tar.bz2
```

Question b:
Answer:
First I moved into my personal bitbucket repository using:

```
cd ~/bitbucket/2016_spring_ecpe170
```

After that I made a new folder for exam1 using:

```
mkdir exam1
```

I then moved the file I downloaded into the previous step into that directory using:

```
cp exam1.tar.bz2 ~/bitbucket/2016_spring_ecpe170/exam1
```

I then extracted the .tar.bz2 file so I could view the source code.

Question c:
Answer:
First I created a new text document and used MakeFile4 from Lab 3 as a base for my new makefile.
All changes for this portion are in the makefile.
I added GCC, used the C99 language standard flag, compiler optimizations, compiler warnings, and added the correct header and object files, I also changed the output binary name to exam1.

Question d:
Answer:
First I went into my exam1 directory:

```
cd ~/bitbucket/2016_spring_ecpe170/exam1
```

I then ran make to compile the program simply using:

```
make
```

After that I ran the program using the command:

```
./exam1
```

Question e:
Answer:
In order to measure the execution time I used the command:

```
time ./exam1
```

The result of this command included:

```
real    0m0.001s
user    0m0.000s
sys     0m0.000s
```

The 'real' time is the best representation of “wall clock” time.

Question f:
Answer:
In order to run Valgrind to get a report of the memory leak in the program I ran this command:

```
valgrind --tool=memcheck --leak-check=yes --show-reachable=yes --num-callers=20 --log-file=memcheck.txt ./exam1
```

I then opened the file created by running Valgrind to see the text report:

```
gedit memcheck.txt &
```

This is the Valgrind output with the memory leak:

```
==3088== Memcheck, a memory error detector
==3088== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==3088== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
==3088== Command: ./exam1
==3088== Parent PID: 2552
==3088==
==3088==
==3088== HEAP SUMMARY:
==3088==     in use at exit: 10,240 bytes in 10 blocks
==3088==   total heap usage: 21 allocs, 11 frees, 10,968 bytes allocated
==3088==
==3088== 10,240 bytes in 10 blocks are definitely lost in loss record 1 of 1
==3088==    at 0x4C2CC70: calloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==3088==    by 0x4008B1: main (main.c:51)
==3088==
==3088== LEAK SUMMARY:
==3088==    definitely lost: 10,240 bytes in 10 blocks
==3088==    indirectly lost: 0 bytes in 0 blocks
==3088==      possibly lost: 0 bytes in 0 blocks
==3088==    still reachable: 0 bytes in 0 blocks
==3088==         suppressed: 0 bytes in 0 blocks
==3088==
==3088== For counts of detected and suppressed errors, rerun with: -v
==3088== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

Question g:
Answer:
The memory leak is created at line 51 since Valgrind reported it, so this is the problem:
“line = calloc(1, line_len);”
I think I should use something like “free(line)”. This compiles, but does not fix the memory leak

This is the Valgrind output after the free(line) is added:

```
==3401== Memcheck, a memory error detector
==3401== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==3401== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
==3401== Command: ./exam1
==3401== Parent PID: 2552
==3401==
==3401== Invalid read of size 1
==3401==    at 0x4E82A03: vfprintf (vfprintf.c:1661)
==3401==    by 0x4E8B498: printf (printf.c:33)
==3401==    by 0x400A0C: line_print (main.c:127)
==3401==    by 0x400938: main (main.c:72)
==3401==  Address 0x51febd0 is 0 bytes inside a block of size 1,024 free'd
==3401==    at 0x4C2BDEC: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==3401==    by 0x400916: main (main.c:65)
==3401==
==3401==
==3401== HEAP SUMMARY:
==3401==     in use at exit: 9,216 bytes in 9 blocks
==3401==   total heap usage: 21 allocs, 12 frees, 10,968 bytes allocated
==3401==
==3401== 9,216 bytes in 9 blocks are definitely lost in loss record 1 of 1
==3401==    at 0x4C2CC70: calloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==3401==    by 0x4008B1: main (main.c:51)
==3401==
==3401== LEAK SUMMARY:
==3401==    definitely lost: 9,216 bytes in 9 blocks
==3401==    indirectly lost: 0 bytes in 0 blocks
==3401==      possibly lost: 0 bytes in 0 blocks
==3401==    still reachable: 0 bytes in 0 blocks
==3401==         suppressed: 0 bytes in 0 blocks
==3401==
==3401== For counts of detected and suppressed errors, rerun with: -v
==3401== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```