

CSP 301 Report of Group 5 for PART A

– Abhinav Deep Singh, 2012CS10207
– Chirag Agrawal, 2012CS50285
– Shivam Garg, 2012CS10252

We ran Valgrind utility on our code and following output came. The analysis of the report is given thereafter.

```
**-----VALGRIND OUTPUT-----
**-----BEGIN-----

cs1120207@varali:~/simulatioc$ valgrind --leak-check=full ./a.out -y 1 >fil.txt
==22751== Memcheck, a memory error detector
==22751== Copyright (C) 2002-2011, and GNU GPL'd, by Julian Seward et al.
==22751== Using Valgrind-3.7.0 and LibVEX; rerun with -h for copyright info
==22751== Command: ./a.out -y 1
==22751==
==22751== Syscall param msgsnd(msgp->mtext) points to uninitialised byte(s)
==22751==    at 0x59573C3: __msgsnd_nocancel (syscall-template.S:82)
==22751==    by 0x421977: send_msg(int, mymsgbuf*) (in /home/btech/cs1120207/simulatioc/a.out)
==22751==    by 0x40248A: main (in /home/btech/cs1120207/simulatioc/a.out)
==22751== Address 0x7ff0003d0 is on thread 1's stack
==22751==
==22751== Syscall param msgsnd(msgp->mtext) points to uninitialised byte(s)
==22751==    at 0x59573C3: __msgsnd_nocancel (syscall-template.S:82)
==22751==    by 0x421977: send_msg(int, mymsgbuf*) (in /home/btech/cs1120207/simulatioc/a.out)
==22751==    by 0x4025F4: main (in /home/btech/cs1120207/simulatioc/a.out)
==22751== Address 0x7ff0003d0 is on thread 1's stack
==22751==
==22756== Thread 5:
==22756== Syscall param msgsnd(msgp->mtext) points to uninitialised byte(s)
==22756==    at 0x59573E3: ??? (syscall-template.S:82)
==22756==    by 0x421977: send_msg(int, mymsgbuf*) (in /home/btech/cs1120207/simulatioc/a.out)
==22756==    by 0x412798: generateCourses(void*) (in /home/btech/cs1120207/simulatioc/a.out)
==22756==    by 0x4E39E99: start_thread (pthread_create.c:308)
==22756==    by 0x59553FC: clone (clone.S:112)
==22756== Address 0x8023e90 is on thread 5's stack
==22756==
==22756== Thread 6:
==22756== Syscall param msgsnd(msgp->mtext) points to uninitialised byte(s)
==22756==    at 0x59573E3: ??? (syscall-template.S:82)
==22756==    by 0x421977: send_msg(int, mymsgbuf*) (in /home/btech/cs1120207/simulatioc/a.out)
==22756==    by 0x412913: genFen(void*) (in /home/btech/cs1120207/simulatioc/a.out)
==22756==    by 0x4E39E99: start_thread (pthread_create.c:308)
==22756==    by 0x59553FC: clone (clone.S:112)
==22756== Address 0x8824e80 is on thread 6's stack
==22756==
==22751==
```

```

==22751== HEAP SUMMARY:
==22751==      in use at exit: 41,480 bytes in 217 blocks
==22751==    total heap usage: 72,573 allocs, 72,356 frees, 564,132 bytes allocated
==22751==
==22751== 4,896 bytes in 34 blocks are definitely lost in loss record 4 of 5
==22751==    at 0x4C2B1C7: operator new(unsigned long) (in /usr/lib/valgrind/vgpreload_memcheck-amd64-l
==22751==    by 0x413BB6: setEnviro() (in /home/btech/cs1120207/simulatioc/a.out)
==22751==    by 0x402205: main (in /home/btech/cs1120207/simulatioc/a.out)
==22751==
==22751== 36,000 bytes in 180 blocks are definitely lost in loss record 5 of 5
==22751==    at 0x4C2B1C7: operator new(unsigned long) (in /usr/lib/valgrind/vgpreload_memcheck-amd64-l
==22751==    by 0x41389E: setEnviro() (in /home/btech/cs1120207/simulatioc/a.out)
==22751==    by 0x402205: main (in /home/btech/cs1120207/simulatioc/a.out)
==22751==
==22751== LEAK SUMMARY:
==22751==    definitely lost: 40,896 bytes in 214 blocks
==22751==    indirectly lost: 0 bytes in 0 blocks
==22751==    possibly lost: 0 bytes in 0 blocks
==22751==    still reachable: 584 bytes in 3 blocks
==22751==    suppressed: 0 bytes in 0 blocks
==22751== Reachable blocks (those to which a pointer was found) are not shown.
==22751== To see them, rerun with: --leak-check=full --show-reachable=yes
==22751==
==22751== For counts of detected and suppressed errors, rerun with: -v
==22751== Use --track-origins=yes to see where uninitialised values come from
==22751== ERROR SUMMARY: 6 errors from 4 contexts (suppressed: 2 from 2)
cs1120207@varali:~/simulatioc$ ==22756==
==22756== HEAP SUMMARY:
==22756==      in use at exit: 568 bytes in 1 blocks
==22756==    total heap usage: 117,370 allocs, 117,369 frees, 1,407,834 bytes allocated
==22756==
==22756== LEAK SUMMARY:
==22756==    definitely lost: 0 bytes in 0 blocks
==22756==    indirectly lost: 0 bytes in 0 blocks
==22756==    possibly lost: 0 bytes in 0 blocks
==22756==    still reachable: 568 bytes in 1 blocks
==22756==    suppressed: 0 bytes in 0 blocks
==22756== Reachable blocks (those to which a pointer was found) are not shown.
==22756== To see them, rerun with: --leak-check=full --show-reachable=yes
==22756==
==22756== For counts of detected and suppressed errors, rerun with: -v
==22756== Use --track-origins=yes to see where uninitialised values come from
==22756== ERROR SUMMARY: 0 errors from 0 contexts
**-----END-----

```

% time	Cumulative Seconds	Self Seconds	Calls	Self Ts/call	Total Ts/call	Name
0.00	0.00	0.00	5282087	0.00	0.00	__gnu_cxx::__normal_iterator<student**, std::vector<student*, std::allocator<student*> > >::base() const
0.00	0.00	0.00	5270728	0.00	0.00	u_cxx::__normal_iterator<student*, std::vector<student*, std::allocator<student*> > >::__normal_iterator(student** const&)
0.00	0.00	0.00	2643204	0.00	0.00	__gnu_cxx::__normal_iterator<student**, std::vector<student*, std::allocator<student*> > >::operator*() const
0.00	0.00	0.00	2622446	0.00	0.00	std::vector<student*, std::allocator<student*> >::end()

[?]

Figure 1: Profiler Part A Output

```

**-----VALGRIND REPORT OF PART A & ANALYSIS-----
**-----BEGIN-----
The report shows that there was no leaks of "definitely lost", "indirectly lost" or "possibly lost" type.

The heap summary result does not change.
There are no errors in the valgrind report.

This report remains same even if no of years are changed.

**-----END-----
//Same code was tested with profiler.
**-----PROFILER REPORT OF PART A & ANALYSIS-----
**-----BEGIN-----
1. All operations take very less time. They don't add up even in cumulative time. Even though the operations are many.
**-----END-----

```