Debugging like the Pros

3157 Hackathon | 10/9/15
We are generating 134 characters.

Our generated character array is:
afzoltgtdddckfxcemgdibgpcxppxwqnhkdpypnfhlnpcqzyzzynbyfxbijnlvjokhxomecemwengmehfjaihblh
thdjtjmiianmyfsnesrxxetfoaixiurebup

HEAP SUMMARY:
in use at exit: 138 bytes in 2 blocks

4 bytes in 1 blocks are definitely lost in loss record 1 of 2
at 0x4C2B1C7: operator new(unsigned long) (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
by 0x400927: main

134 bytes in 1 blocks are definitely lost in loss record 2 of 2
at 0x4C2AC27: operator new[](unsigned long) (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
by 0x40099F: main

LEAK SUMMARY:
definitely lost: 138 bytes in 2 blocks
indirectly lost: 0 bytes in 0 blocks
possibly lost: 0 bytes in 0 blocks
still reachable: 0 bytes in 0 blocks
suppressed: 0 bytes in 0 blocks
We are generating 134 characters
Our generated character array is:

We are getting 134 characters
Our generated character array is:

4 bytes in 1 blocks are definitely lost record 1 of 2
at 0x4C2B1C7: operator new (/usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
by 0x400927: main

134 bytes in 1 blocks are definitely lost record 2 of 2
at 0x4C2AC27: operator new (/usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
by 0x40099F: main

LEAK SUMMARY:
definitely lost: 138 bytes in 2 blocks
indirectly lost: 0 bytes in 0 blocks
possibly lost: 0 bytes in 0 blocks
still reachable: 0 bytes in 0 blocks
suppressed: 0 bytes in 0 blocks
TIP #1

Use printf()
PRINTFO
ALL THE THINGS
Use `printf()`
Use `printf()`

- Check yourself: what values are *actually* being passed into / returned from functions?
Use `printf()`

- Check yourself: what values are *actually* being passed into / returned from functions?
- You should know the value of all variables at all times!
TIP #2

Use `fprintf(stderr)`
Use fprintf(stderr)
Use `fprintf(stderr)`

- `stdin` is buffered, so output may not be displayed immediately (or at all!)
Use `fprintf(stderr)`

- `stdin` is buffered, so output may not be displayed immediately (or at all!)
- `stderr` is not buffered, so all statements are outputted immediately
TIP #3

// is your friend
COMMENT
ALL THE THINGS!!
// is your friend
// is your friend

• Multiple errors can be cause by one bug!
// is your friend

- Multiple errors can be cause by one bug!
- Address the first error first
TIP #4

DRAW A PICTURE!!!
DRAW A PICTURE!!!
• Map out memory, pointers, and variables
DRAW A PICTURE!!!

- Map out memory, pointers, and variables
- Follow the data!
TIP #5

Errors cause each other
Errors cause each other
Errors cause each other

• Multiple errors can be cause by one bug!
Errors cause each other

• Multiple errors can be caused by one bug!
• Address the first error first
TIP #6

Become a Valgrind Detective
Become a Valgrind Detective
Become a Valgrind Detective

• All those errors? They’re actually clues!
Become a Valgrind Detective

• All those errors? They’re actually clues!
• Line numbers
Become a Valgrind Detective

• All those errors? They’re actually clues!
• Line numbers
• Error text
Become a Valgrind Detective

• All those errors? They’re actually clues!
  • Line numbers
  • Error text
  • Number of errors
TIP #7

Test early, test often.
I DON'T TEST OFTEN

BUT WHEN I DO, I GET 1,000 GCC ERRORS
Test early, test often.
Test early, test often.

• No one wants to be overwhelmed with GCC errors or memory leaks
Test early, test often.

• No one wants to be overwhelmed with GCC errors or memory leaks

• Code incrementally
We are generating 134 characters

Our generated character array is:

```
afzoltgtdcetakxmdegdpwphcfppqyzznbyfsxbjnlvjkxhcmecwqemkhefjaih[
```

**HEAP SUMMARY:**
- in use at exit: 138 bytes in 2 blocks
- total heap usage: 2 allocs, 0 frees, 138 bytes allocated

4 bytes in 1 blocks are definitely lost in loss record 1 of 2
- at 0x4C2B1C7: operator new(unsigned long) (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
- by 0x400927: main

134 bytes in 1 blocks are definitely lost in loss record 2 of 2
- at 0x4C2AC27: operator new[](unsigned long) (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
- by 0x40099F: main

**LEAK SUMMARY:**
- definitely lost: 138 bytes in 2 blocks
- indirectly lost: 0 bytes in 0 blocks
- possibly lost: 0 bytes in 0 blocks
- suppressed: 0 bytes in 0 blocks
We are generating 134 characters
Our generated character array is:

```
amaflzoltgtdecfxilcmegdebipgypcpxppxwqnhkwdpypycffhnnlcpgqgzzyzznbyfsxbjnlvbjokhxomecwcqengmehkefhfjaih
```

HEAP SUMMARY:
in use at exit: 138 bytes in 2 blocks
  total heap usage: 2 allocs, 0 frees; 138 bytes allocated

4 bytes in 1 blocks are definitely lost, record 1 of 2
at 0x4C2B1C7: operator new[] (in /usr/lib/valgrind/vgpreload_memcheck amd64-linux.so)
  by 0x400927: main

134 bytes in 1 blocks are definitely lost, record 2 of 2
at 0x4C2AC27: operator new[] (in /usr/lib/valgrind/vgpreload_memcheck amd64-linux.so)
  by 0x40099F: main

LEAK SUMMARY:
definitely lost: 138 bytes in 2 blocks
indirectly lost: 0 bytes in 0 blocks
possibly lost: 0 bytes in 0 blocks
still reachable: 0 bytes in 0 blocks
suppressed: 0 bytes in 0 blocks