debugging/valgrind/example_use_after_free

Valgrind

Example: Valgrind - Use after free

- Source code `use_after_free.c++`

  ```
  #include <iostream>

  int main(int argc, char *argv[]) {
      auto *charArray = new char[10];
      delete [] charArray;
      std::cout << charArray[0] << std::endl;
  }
  ```

- Set up valgrind and build environment

  ```
  module add \
      compiler/gnu \
      devel/valgrind
  ```

- Build application

  ```
  c++ -O1 -g use_after_free.c++ -o use_after_free
  ```

- Examine `use_after_free` program with valgrind memchecker

  ```
  valgrind ./use_after_free
  ```

  ```
  ==3318937== Memcheck, a memory error detector
  ==3318937== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
  ==3318937== Using Valgrind-3.20.0 and LibVEX; rerun with -h for copyright info
  ==3318937== Command: ./use_after_free
  ==3318937==
  ==3318937== Invalid read of size 1
  ==3318937==    at 0x400840: main (use_after_free.c++:6)
  ==3318937==  Address 0x5bcec80 is 0 bytes inside a block of size 10 free'd
  ==3318937==    at 0x4C3B6FB: operator delete[](void*) (in /.../libexec/valgrind/vgprelo
  ==3318937==    by 0x40083F: main (use_after_free.c++:5)
  ==3318937==  Block was alloc'd at
  ```

```
==3318937==    at 0x4C391AF: operator new[](unsigned long) (in /.../libexec/valgrind/vg
==3318937==    by 0x400834: main (use_after_free.c++:4)
==3318937==

==3318937==
==3318937== HEAP SUMMARY:
==3318937==     in use at exit: 0 bytes in 0 blocks
==3318937==   total heap usage: 3 allocs, 3 frees, 73,738 bytes allocated
==3318937==
==3318937== All heap blocks were freed -- no leaks are possible
==3318937==
==3318937== For lists of detected and suppressed errors, rerun with: -s
==3318937== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```