CS100 Introduction to Programming

Recitation 10

<Yang Feiming>

<yangfm@shanghaitech.edu.cn>

NO PLAGIARISM!!!

- The most likely cause for failing this course.
- You WILL be caught!
- We WILL punish!
- They WILL know!
  - Parents
  - University
  - School
  - Fellows

Clone the skeleton code

- `git clone https://github.com/llk89/cs100recitation10`

- This will be used in most part of the recitation

Overview

- Move in containers
- GUI assisted debugging
- Profiling in action

Move in containers

Move in STL

- Most containers implements move semantic
- Why?

Create a vector with complex content

- In `move/Month.cpp`
- Key points:
  - Pre-allocate: `reserve()`
  - Call constructors with moved arguments: `emplace_back()`

```
std::vector<Month> listOfMonths() {
    std::vector<Month> months;
    const size_t months_count = sizeof(month_names) / sizeof(*month_names);
    months.reserve(months_count);
    for (int i = 0; i < months_count; ++i) {
        months.emplace_back(i, std::string(month_names[i]));
    }
    return months;
}
```

Drain a vector in reversal

- In move/main.cpp
- Fix the inefficiency in this function

```
template<typename T>
void drain_reversed(vector<T>& to, vector<T> &&from) {
    cout << "draining" << endl;
    while (!from.empty()) {
        to.push_back(from.back());
        from.pop_back();
    }
    cout << "drained" << endl;
}
```

Drain a vector in reversal

- In move/main.cpp
- Can we use reference instead of rvalue-reference here?

```
template<typename T>
void drain_reversed(vector<T>& to, vector<T> &&from) {
    cout << "draining" << endl;
    while (!from.empty()) {
        to.push_back(from.back());
        from.pop_back();
    }
    cout << "drained" << endl;
}
```

Combine it all

- Now compile and run this program.
- Observe the peculiar output.
  - Why there are a bunch of `destruct X X` in the output?
  - Why `move` and `destruct` appears in alternation?
- How many copies of January exists throughout the entire execution?
  - How many if we didn’t move in drain_reverse?

GUI assisted debugging

Why GUI?

- May be easier
- Show current line
- Display all locals automatically
- ……
  - Still depends on your personality

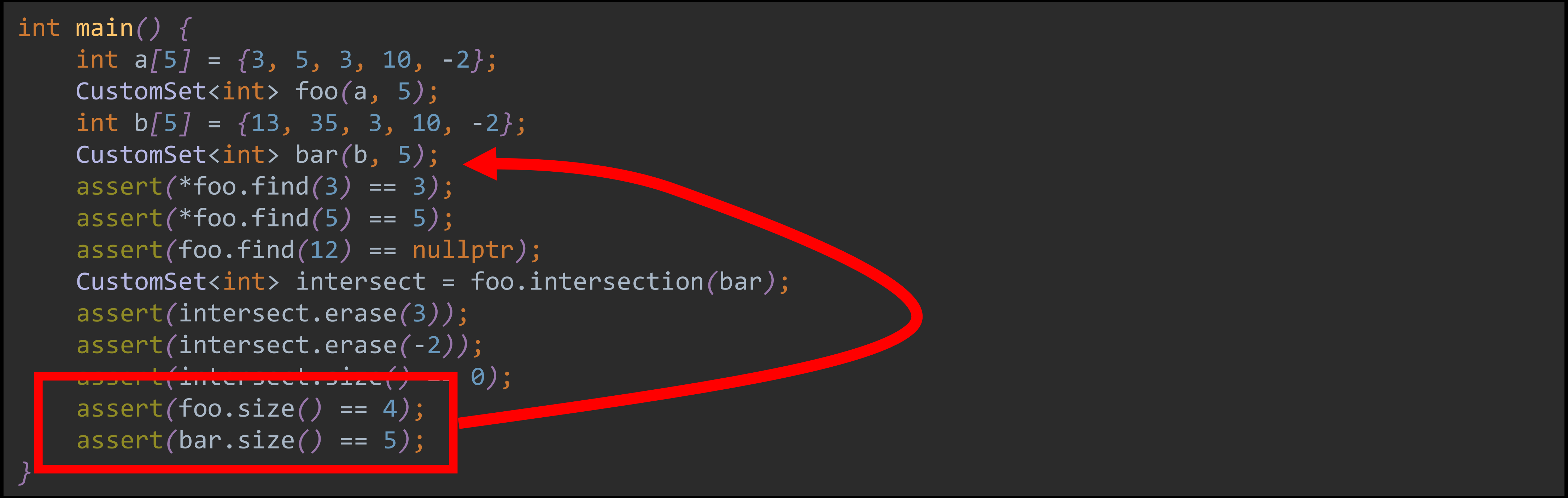
Living debugging session

- In buggy/
- There are two problems within.
- The test driver would fails every assertion.

Bug hunt

Aftermath

- Test driver?
  - Test basic functionality first



```cpp
int main() {
    int a[5] = {3, 5, 3, 10, -2};
    CustomSet<int> foo(a, 5);
    int b[5] = {13, 35, 3, 10, -2};
    CustomSet<int> bar(b, 5);
    assert(*foo.find(3) == 3);
    assert(*foo.find(5) == 5);
    assert(foo.find(12) == nullptr);
    CustomSet<int> intersect = foo.intersection(bar);
    assert(intersect.erase(3));
    assert(intersect.erase(-2));
    assert(intersect.size() == 0);
    assert(foo.size() == 4);
    assert(bar.size() == 5);
}
```

Profiling in action

How to solve a linear system *fast*

- Recall the last pages of lecture 19
- SVD, QR, NORMAL, which fastest?
- Implement a LapTimer or some sort and design a micro benchmark

- You can take the code from lecture slides

Compiler Optimizations

- gcc -O3?

Importance of test data set

- Sparse or dense?
- Size of matrix?
- Already triangular?
- ……

Parallel?

- Parallel libraries?
  - MKL
  - MPI
  - OpenACC
  - OpenMP
- GPU?

QA Time

- If you have any problems with…
  - Homework 6
  - Lectures
  - Recitation 10
- Ask now