Midterm Review

Computer Systems Programming, Spring 2025

Instructor: Travis McGaha

Teaching Assistants:

Andrew Lukashchuk	Ashwin Alaparthi	Lobi Zhao
Angie Cao	Austin Lin	Pearl Liu
Aniket Ghorpade	Hassan Rizwan	Perrie Quek

Poll Everywhere

pollev.com/tqm

- How are you? What questions do you have about pipe?

Administrivia

- Midterm this week ☺
  - Midterm review in recitation last week
    - No recitation this week
  - Midterm review in lecture **TODAY**
  - Policies and old exam posted

- "Check-in" posted
  - Due Friday

Midterm Philosophy / Advice (pt. 1)

- I do not like midterms that ask you to memorize things
  - You will still have to memorize some critical things.
  - I will hint at some things, provide documentation or a summary of some things. (for example: I will list some of the functions that may be useful and a brief summary of what the function does)

- I am more interested in questions that ask you to:
  - Apply concepts to solve new problems
  - Analyze situations to see how concepts from lecture apply

- Will there be multiple choice?
  - If there is, you will still have to justify your choices

Midterm Philosophy / Advice (pt. 2)

- I am still trying to keep the exam fair to you, you must remember some things
  - High level concepts or fundamentals. I do not expect you to remember every minute detail.
    - E.g. how a multi level page table works should be know, but not the exact details of what is in each page table entry
    - (I know this boundary is blurry, but hopefully this statement helps)
- I am NOT trying to “trick” you (like I sometimes do in poll everywhere questions)

Midterm Philosophy / Advice (pt. 3)

- I am trying to make sure you have adequate time to stop and think about the questions.
  - You should still be wary of how much time you have
  - But also, remember that sometimes you can stop and take a deep breath.

- Remember that you can move on to another problem.

- Remember that you can still move on to the next part even if you haven't finished the current part

Midterm Philosophy / Advice (pt. 4)

- On the midterm you will have to explain things
- Your explanations should be more than just stating a topic name.
- Don't just say something like (for example) "because of threads" or just state some facts like "threads are parallel and lightweight processes".
- State how the topic(s) relate to the exam problem and answer the question being asked.

Disclaimer

- **THIS REVIEW IS NOT EXHAUSTIVE**
- **Topics not in this review are still testable**
  - **We recommend going through the course material. Lecture polls, recitation worksheets, and the previous homeworks.**

Review Topics

- C++ Programming
- C++ Memory
- Processes
- Caches

C++ Programming (pt 1)

- Implement the function filter() which takes in a vector of integers and a set of integers. The function returns a new vector that contains all of the integers of the input vector, except for any elements that were in the set.

- For example, the following code should print
  - 4
  - 5

```
vector<int> v {3, 4, 5};
set<int> s {3, 6};

auto res = filter(v, s);

for (auto& num : res) {
  cout << num << endl;
}
```

C++ Programming (pt 1)

```
vector<int> filter(const vector<int>& numbers
                   const set<int>& omit) {

  vector<int> result{};

  for (const auto& num : numbers) {
    if (!omit.contains(num)) {
      result.push_back(num);
    }
  }
  return result;
}
```

C++ Programming (pt 2)

- Implement the function invert() which takes in a map that maps strings to other strings. The function returns a map of strings to vectors of strings that represents the "reverse mapping" of the input map. In other words, the keys in the result map should be all the values in the input map. The values in the output map should be all keys that mapped to that value in the input map.

- For example, consider:

```
map<string, string> map;
map["radar"] = "tacoma";
map["rain"] = "tacoma";
map["transit"] = "philly";

map<string, vector<string>> res = invert(map);
// res should be:
// {
//   "tacoma" -> ["radar", "rain"],
//   "philly" -> ["transit"],
// }
```

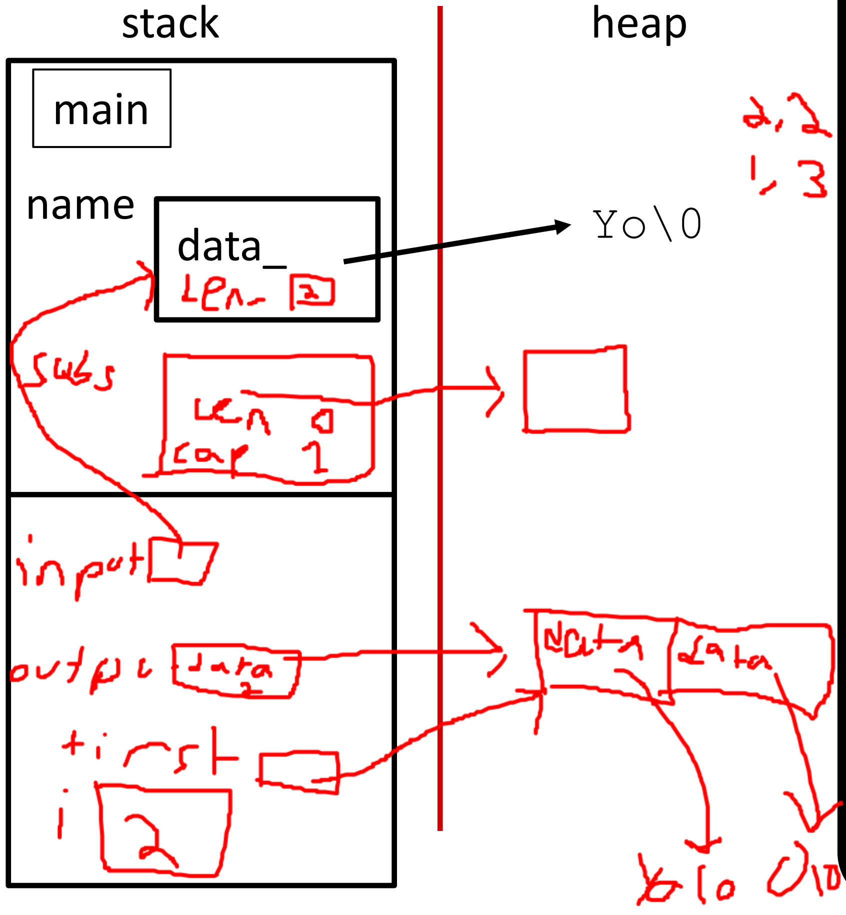
C++ Programming (pt 1)

```
map<string, vector<string>> invert(const map<string, string>& map) {
  map<string, vector<string>> res;

  for (const auto& kv : map) {
    res[kv.first].push_back(kv.second);
  }

  return res;
}
```

C++ Memory

11

- Draw the state of memory when the code reaches // HERE
  We give a little to start.



```
vec<strings>
~~void~~ make_subs(const string& input, vector<string> output) {
  size_t i = 0;
  while (i < input.size()) {
    string sub = input.substr(i);
    output.push_back(sub);
    i += 1;
  }
  string& first = output.at(0);
  // HERE
  return output;
}

int main() {
  string name = "Yo";
  vector<string> subs; // assume initial capacity of 1
  subs = make_subs(name, subs);
}
```

Yo

o

C++ Memory

- How many memory allocations occur in this code?

- 11
  see recording for explanations for this and the previous slide.

```
vector<string> make_subs(const string& input, vector<string> output) {
  size_t i = 0;
  while (i < input.size()) {
    string sub = input.substr(i);
    output.push_back(sub);
    i += 1;
  }
  string& first = output.at(0);
  // HERE
  return output;
}

int main() {
  string name = "Yo";
  vector<string> subs; // assume initial capacity of 1
  subs = make_subs(name, subs);
}
```

Processes

- Consider this code,
  what is one possible output?

- What changes do we need to make so that each child process prints exactly one number?
  - We still need to use processes fork, I and the while loop. You may not delete existing lines of code, but you can add new lines or add onto existing lines of code. The same number of processes should be created as before.

```
int i = 0;

int main() {
  while (i < 2) {
    pid_t pid = fork();

    cout << i;

    i += 1;
  }
}
```

Processes

- Consider this code,
  what is one possible output?
  - **0 1 0 1 0 1 0 1**
  - **There are four processes in total**
  - **Each process prints 0 and 1. The reason why some processes print 0 even if they didn't execute cout << 0; is because the 0 is still in the cout buffer when we fork.**

```
int i = 0;

int main() {
  while (i < 2) {
    pid_t pid = fork();

    cout << i;

    i += 1;
  }
}
```

Processes

- What changes do we need to make so that each child process prints exactly one number?
  - We still need to use processes fork, I and the while loop. You may not delete existing lines of code, but you can add new lines or add onto existing lines of code.
  - **We need to make the print conditional and flush the buffer when we print**

```
int i = 0;

int main() {
  while (i < 2) {
    pid_t pid = fork();
    if (pid == 0) {
      cout << i << endl;
    }
    i += 1;
  }
}
```

Caches Q1

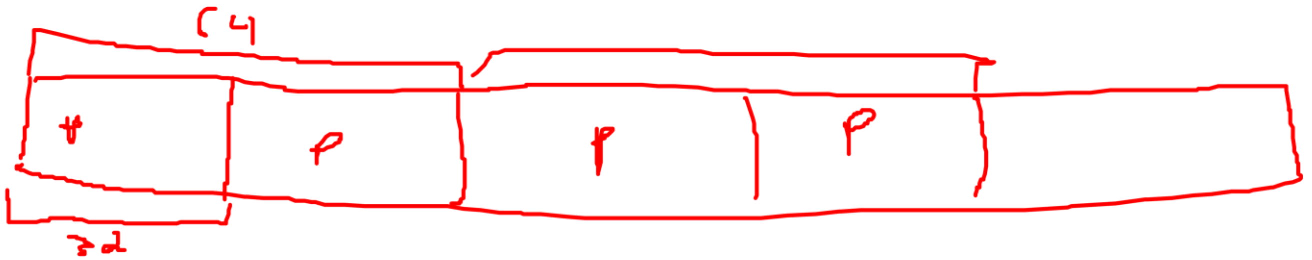
- Let's say we are making a program that simulates various particles interacting with each other. To do this we have the following structs to represent a color and a point

```
struct color {
  int red, green, blue;
};
```

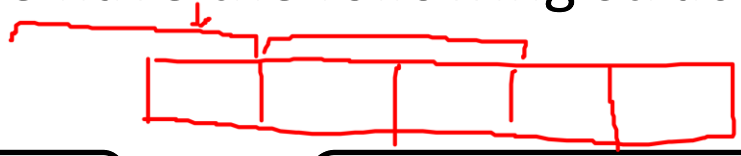
```
struct point {
  double x, y;
  struct color c;
};
```

- If we were to store 100 point structs in an array, and iterate over all of them, accessing them in order, roughly how many cache hits and cache misses would we have?
  - Assume:
    - a cache line is 64 bytes
    - the cache starts empty
    - `sizeof(point)` is 32 bytes, `sizeof(color)` is 16 bytes

Caches Q1



- Let's say we are making a program that simulates various particles interacting with each other. To do this we have the following structs to represent a color and a point



```
struct color {
  int red, green, blue;
};
```

```
struct point {
  double x, y;
  struct color c;
};
```

- If we were to store 100 point structs in an array, and iterate over all of them, accessing them in order, roughly how many cache hits and cache misses would we have?
  - Assume:
    - a cache line is 64 bytes
    - the cache starts empty
    - `sizeof(point)` is 32 bytes, `sizeof(color)` is 16 bytes

Roughly every other time we access a point struct, it will already be in the cache. The other 50% of the time, it needs to be fetched from memory

Caches Q2

- Consider the previous problem with point and color structs.
- In our simulator, it turns out a VERY common operation is to iterate over all points and do calculations with their X and Y values.
- How else can we store/represent the point objects to make this operation faster while still maintaining the same data? Roughly how many cache hits would we get from this updated code?

Caches Q2

- Consider the previous problem with point and color structs.
- In our simulator, it turns out a VERY common operation is to iterate over all points and do calculations with their X and Y values.
- How else can we store/represent the point objects to make this operation faster while still maintaining the same data? Roughly how many cache hits would we get from this updated code?

Change point to just be:

```
struct point {
  double x, y;
}
```

Then Store two arrays:

```
array<point, 100> arr1;
array<color, 100> arr2;
// point at index I
// has color arr2[i]
```

Each time we access a point, we can now load 4 points into the cache. We now get ~25 cache misses and 75 hits