Buffers and Assignment 1 Info

CPSC 457 Week 2

Jesse Francis

January 28, 2020

Buffers

Why do we use buffers in programming?

We use buffers because often writing data to the buffer is faster than immediately operating on it.

How Do We Use Buffers?

We might originally have something like this:

```c
#include <unistd.h>
const int BUFSZ = 10000; // the amount of 'a's to process

void process_data(const void * data, int data_length) {
  usleep(10); // function startup cost
  usleep(data_length);
  usleep(10); // function shutdown cost
}

int main() {
  char buf = 'a'; // declare a 'buffer' of size 1 holding an 'a'
  for (int i = 0; i < BUFSZ; i++) {
    // process the 'a' BUFSZ times
    process_data(&buf, 1);
  }
}
```

Performance of **processNoBuffer.cpp**

```
real    0m2.267s
user    0m0.045s
sys     0m0.131s
```

An Improvement Using A Buffer

We can improve our performance

```c
#include <unistd.h>
const int BUFSZ = 10000; // the amount of 'a's to prcoess

void process_data(const void * data, int data_length) {
  usleep(10); // function startup cost
  usleep(data_length);
  usleep(10); // function shutdown cost
}

int main() {
  char buf[BUFSZ]; // declare a buffer of size BUFSZ
  for (int i = 0; i < BUFSZ; i++) {
    buf[i] = 'a'; // fill the buffer with 'a's
  }
  // process all the data at once
  process_data(&buf, BUFSZ);
}
```

Performance of **processWithBuffer.cpp**

```
real    0m0.014s
user    0m0.001s
sys     0m0.001s
```

Assignment Info

Email Availability

- I will be answering questions that get into my inbox by 11:59pm Thursday. If you email me Friday with questions about Assignment 1, you will not get a reply.

General Reminders

- The assignment is due this Friday.
- Don't plagiarize. If you can find a solution online, I can too.
    - The Midterm and Final

Question 5

- Q5 of the assignment must run on the Linux computers and produce the same output as the given `countLines.cpp`.
  - I will be testing your program with a file of unknown (to you) size. So don't hard-code the number of lines or make assumptions about the size of the file.
  - The number of newlines in the file will still fit within an `int`.
- You will get 0 marks for using fstreams or C++ streams instead of `read()` to read the file.
  - Using C++ streams (`std::cout`) for output to `stdout` is allowed and recommended.
- If you want to know more about the read command, use:
  `man read.3`
- Please don't make your code unreadable. If your code is not well styled or completely lacks comments, you will lose 1 mark.

Exercises

Exercises

1. Try running the buffer examples on your own.
2. Work on your assignment.