ECE454 Lab 2

2019 – Jack Luo
Lab2 – Goals

• Practice program optimization workflow

• Gain understanding & practical experience in
  • Optimizing program for memory performance
  • Using profiling tools + inspection tools
  • Optimizations performed by the compiler
  • Man page reading comprehension and googling skills
Lab2 – Overview

• **Time Frame**: Sept 19 – Oct 13 (~3 week)

• **Lab Marks**: Worth 9% of total course mark
  - 30% competitive, 70% non-competitive (cutoff is 1\textsuperscript{st} week average)

• **Team Composition**: Individual work

• **My Suggestion**:
  
  Do not perform pre-mature program optimizations
  
  Do spend lots of time on what compiler is doing behind your back
  - Pretend you are the compiler and optimize for algorithm + code
  
  Do spend more time reading, thinking, profiling rather than writing code
  - “Genius is 2% inspiration, 98% perspiration” - Thomas Edison
Lab2 – Overview

Start with a square image

- White pixel = background
- Non-white pixel = object
Lab2 – Overview

Given a series of inputs

- WASD, Rotate CW+CCW, Mirror X+Y
- Manipulate object inside the image
- Output result image every 25 input instructions

Move down by 10 pixels
Lab2 – Overview

Given a series of inputs

- WASD, Rotate CW+CCW, Mirror X+Y
- Manipulate object inside the image
- Output result image every 25 input instructions
Lab2 – Overview

Given a series of inputs

• WASD, Rotate CW+CCW, Mirror X+Y
• Manipulate object inside the image
• Output result image every 25 input instructions
Lab2 – Overview

Given a series of inputs

- WASD, Rotate CW+CCW, Mirror X+Y
- Manipulate object inside the image
- Output result image every 25 input instructions
Lab2 – Optimization Headroom

I have tried very hard to make the reference solution slow

Level of difficulty:

- You should easily achieve 250x+ speedup with modest effort in the right places
- Top solution from last year achieved 7000x+ speedup with this year’s inputs
- One of this year’s new TA got around 1000x speedup in 1 day

Really big hints:

- Think hard on why the reference solution is slow
- Compiler optimization techniques can be used in your algorithms as well
- Don’t perform premature optimization (readability is important)
- Improve your solution and lower other’s competitive marks
Lab2 – Optimization Workflow

1. Evaluate current algorithm
2. Optimize your algorithm
3. Profile your code
4. Optimize for system performance

Give up because
- Your time is better spent elsewhere such as studying midterm
- You are too good and you are afraid others beat you
  - Warning to 1st place student: TA is not responsible for your personal safety
Helpful Resource

GCC Man Page: https://lmgtfy.com/?q=gcc+man+page
Gprof Tutorial: https://lmgtfy.com/?q=gprof+tutorialspoint
Gcov Tutorial: https://lmgtfy.com/?q=gcov+tutorialspoint
Git tutorial: https://lmgtfy.com/?=git+tutorial+for+beginners
Bash Script Tutorial: https://www.shellscript.sh/