"More than the act of testing, the act of designing tests is one of the best bug preventers known. The thinking that must be done to create a useful test can discover and eliminate bugs before they are coded – indeed, test-design thinking can discover and eliminate bugs at every stage in the creation of software, from conception to specification, to design, coding and the rest." – B. Bezier

"The speed of a non-working program is irrelevant." – S. Heller (in "Efficient C/C++ Programming")

Learning Objectives

1. Assembling larger programs from components
2. Concurrency
3. Creative problem solving

Work that needs to be handed in (via git)

This lab is due May 3rd at 8 PM, and only ONE team member should submit.

1. `spimbot.s`, your SPIMbot tournament entry,
2. `partners.txt`, a list of you and your 1 or 2 partners' NetIDs,
3. `writeup.txt`, a few paragraphs (in ASCII) that describe your strategy and any interesting optimizations that you implemented, and
4. `teamname.txt`, a name under which your SPIMbot will compete. Team names must be 40 characters or less and should be able to be easily pronounced. Any team names deemed inappropriate are subject to sanitization and disciplinary action.

Guidelines

- **You must do this assignment in groups of 2 or 3 people.** Teamwork is an essential skill for future courses and in the professional world, so it's good to get some practice. If you do the assignment individually, you won't be entered in the tournament, so you can earn at most 70% of the points for this lab. We may decide to change this policy, please look at piazza for updates.
- Use any MIPS instructions or pseudo-instructions. In fact, anything that runs is fair game (*i.e.*, you are not required to observe calling conventions, but remember calling conventions will aid debugging).
- **We reserve the right to not debug code that does not follow calling conventions.**
- All your code must go in `spimbot.s`.
- We will not try to break your code; we will compete it against the other students.
- syscalls will be disabled for this lab
- The contest will be run on the EWS Linux machines, so those machines should be considered to be the final word on correctness. Be sure to test your code on those machines.
- Refer to the SPIMbot documentation for details on its interfaces:
  `cs233.github.io/assets/pdfs/labs/spimbot_documentation.pdf`
- We may need to update the binary if a gamebreaking bug is found that prevents the tournament from running. We will give you as much time as possible to adapt to binary changes.

Story: (Un)Ethical Hacking

A very evil TA wants to take over the world and rebuild it in her image. With all the chaos in the world, now is the time to strike! She has decided to recruit **you** and many of your peers in CS233 to help. Using your extensive knowledge of MIPS, you will help her hack into computers, or hosts, to build the largest botnet in history. Your bots will surf the world wide web and use their port scanners to find and identify targets. Your bots will have access to the most advanced DDOS scripts available on the black market. To help finance her plans for world domination, she has decided to include bytecoin miners on all the bots. The premier currency of the criminal underground, bytecoins will be the currency of her new world order. MIPS will become the new world language and everyone will be forced to learn datapaths. All will bow down to the Empress!

Unfortunately, the professor has caught on to her plan and has recruited his own team of CS233 students. These students are lame and don't want to have fun. In a bid to stop her, he has decided to assemble his own competing botnet. Thanks to a traitor, he has access to the same technology as the Empress. He claims to be all that is left to preserve the light of liberty. He also wants everyone to learn finite state machines.

The rest of the course staff is indifferent to the struggle.

Who will you back? Who do you want to win? The epic saga that is about to unfold will be told through out the generations. Survival is not guaranteed. So begin writing your bots! Prepare for the struggle! Complete this lab before the deadline! This is not a time for hesitation.

The Game

Objective

The goal of this game is to capture hosts and eliminate the enemy. You get points for both. You have a scanner that tells you what you are facing, and a UDP packet to capture hosts or fire at the enemy. The more hosts you have than the enemy, the greater lead in points you will take. Shooting the enemy causes them to respawn, causing them to retreat to one of their hosts.

Winning

In order to win the competition you should get a higher score than your opponent. This can be achieved, not just by capturing hosts, but by shooting the enemy with UDP packets. Every opponent you eliminate will give you 15 points. Every $250k$ cycles, the game will give you a point for every host you have captured.

There are many ways to optimize your bot to beat your opponent. Here are a few things you may want to consider examining and optimizing if you want to create a highly competitive bot:

- Optimize the puzzle solver. We gave you a fairly terrible puzzle solver. There are many ways to improve the cycle efficiency or big O of the algorithm. Google and your brain are your friends.
- Solving puzzles while your bot is moving to a new location, will allow you to have more bytecoins, and thus be able to use the scanner more and fire more UDP packets.
- Being smart about your scanner. The scanner is a zero width beam. What can you do to have more situational awareness?
- UDP packets have travel time. They will not reach their target immediately.
- You have an opponent. A winning strategy will take this into account.
- We have provided the `-test` flag to give you infinite bytecoins for testing purposes.

Qualifications

In order to qualify for the tournament, you must be able to outperform the baseline bot we provided in `adversary.s`. You must be able to beat the baseline bot by at least a 200 point margin in at least 2 games out of a total 3 games. Each of these 3 games will use a different puzzle and map seed. This is the command we will use to run your code on some seed X for qualifications:

```
QtSpimbot -file spimbot.s -file2 adversary.s -puzzleseed X -mapseed X -pt3
```

Tournament

Once you have qualified locally. You will then have to compete in a tournament against your classmates. We will use the following command to run two different spimbots against eachother:

```
QtSpimbot -file spimbotA.s -file2 spimbotB.s -tournament
```

Grading

You are gauranteed at least 70% for the lab **if you pass the qualifications**. If you don't qualify, we will take the best margin you had across all 3 rounds, and multiply that by $\frac{70}{200}$ as your final grade. Otherwise, the remaining 30% is based on your standing in the tournament. Details for this will be released as we get closer to the deadline. The basic idea however is that your tournament grade is determined by your best placement across 10 runs of the final tournament. Even a basic bot has a chance of advancing quite far into the tournament. It's a lot of fun and it is easy to do well if you put the effort in.

Changes for Lab9

There are two changes main changes since Lab9:

- The respawn mechanic now blacklists hosts too close to the enemy
- UDP packets now have a range of 15 and the scanner has a range of 20

Please read the documentation for more details.

SPIMArena

In previous semesters, we have set up a computer in Siebel to run a live tournament of your bots so that you can compare the performance of your bot to your peers before the deadline. Due to circumstances out of our control, this might not be possible this semester. Watch piazza for updates.

Good Luck!