Final Assignment:

Integration of Tools and Practices

By Mohammad Yavari

Computer Workshop

January 27, 2024

Iran University of Science & Technology

Contents

1 Git and GitHub:

1.1 Repository Initialization and Commits

on GitHub, New Repository, Give it a Name, Clone it into Your PC (e.g. using the `https` link and `git clone` command), Create a `main.tex` file and `.github/workflows/main.yml` (further explained).

1.2 GitHub Actions for LaTeX Compilation

first I did'nt know about the tags so I tried to do it on every push and remove the tag condition but when I discovered the releases that are created automatically and went through some difficulties and problems.
So I learned tags and I'm happy about it!
You simply need to use the command `git tag` after the commit you want to be complied and then push using `git push origin tag-name` (you need `git push` too)
(I did'nt change the `main.yml` file)

2 Exploration Tasks

2.1 Vim Advanced Features

1. **Multiple Cursors with Visual Mode:**
   Vim allows you to use visual mode to create multiple cursors and edit multiple occurrences simultaneously.
   1. Use `Ctrl+v` to enter visual block mode.
   2. Move to select the desired columns or lines.
   3. Press `I` to insert text simultaneously on all selected lines.
   4. Press `Esc` to apply changes.

2. **Folding:**
   Vim supports code folding, allowing you to collapse and expand sections of your code. This is beneficial for navigating large files or focusing on specific parts of the code.
   1. `zf` followed by a `motion`: Create a fold.
   2. `zo` and `zc`: Open and close folds, respectively.
   3. `:set foldmethod=indent`: Automatically fold based on the indentation.

3. **Marks and Jumps:**
   Marks allow you to bookmark a location in a file. You can set a mark with `mletter`, and then jump to that mark with `'letter`.
   1. `ma`: Set mark a at the current cursor position.
   2. `'a`: Jump to the position of mark `a`.

Jumps can be used to quickly navigate between different parts of your file. Vim also has a "jump list" (`:jumps`) that keeps track of recent cursor movements.

2.2 Memory profiling

Normally C stores data in static mode (as I know), it has pros and cons, it's more stable and reliable and also easier but gives us less control, if we want control and flexibility we can use dynamic memory allocation which we can use by functions like `malloc() / calloc() / realloc() / free()`,
the syntax and usage are so simple!

2.2.1 Memory Leak

a memory leak happens when your program forgets to return the borrowed space. It's like borrowing a pen but never giving it back. If you keep doing this, you run out of pens, and your backpack gets heavy with stuff you don't need. in other words, remember to `free()` after allocation!

2.2.2 Memory Profilers

Memory Leak Detection: Valgrind keeps track of every allocation and deallocation of memory during the execution of a program.
If your program forgets to free up memory (a memory leak), Valgrind will point it out. It tells you where the leak occurred and how much memory is involved.

Memory Corruption Detection: Valgrind can detect issues like accessing memory beyond its boundaries, which can lead to unpredictable behavior and crashes.(Like arrays in c)
If your program writes to or reads from memory it shouldn't, Valgrind can catch these errors.
And More...

2.3 GNU/Linux Bash Scripting

2.3.1 fzf

Q1: Fuzzy searching is like having a helpful friend who finds what you're looking for even if you don't know the exact details. It's a search technique that doesn't demand precision. Instead, it accommodates small mistakes, typos, or variations in spelling.
it brings "fuzziness" into the search!

Q2: it takes the list of files and directories from the ls command and allows you to interactively search and select one of them using a fuzzy search algorithm.
when you start typing it starts searching and highlighting a result!
(the output of `ls` is sent to `fzf` as input)

2.3.2   Using fzf to find your favorite PDF

1. `fd -e pdf`
2. `fd -e pdf | fzf`

2.3.3   Opening the file using Zathura

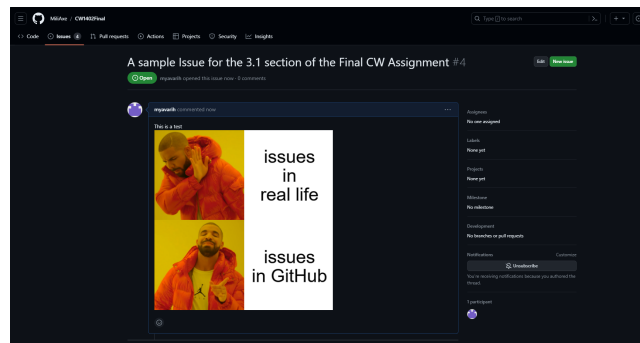`zathura "$(fd -e pdf | fzf)"`

3   Git and FOSS

3.1   Issues

Figure 1: Sample Issue