Final Assignment: Computer Workshop Course

Hazhir Yousefi

January 25, 2025

Contents

1 Git and GitHub

1.1 Repository Initialization and Commits

To start this assignment, I created a new repository on GitHub named ‘latex‘. Below are the steps I followed:

1. Created the repository on GitHub and added a ‘.gitignore‘ file for LaTeX to avoid unnecessary files being tracked.
2. Cloned the repository to my local machine using the command:

   ```
   git clone https://github.com/Hazhir2002/latex
   cd latex
   ```

3. Created a basic LaTeX document named ‘assignment.tex‘ and structured it with sections for each task.
4. Made meaningful commits regularly as I progressed. For instance:

   ```
   git add assignment.tex
   git commit -m "Add initial structure of LaTeX document"
   git push
   ```

1.2 GitHub Actions for LaTeX Compilation

To automate the compilation of my LaTeX document, I set up a GitHub Actions workflow. Below are the steps I followed:

1. Created a new directory in my repository named ‘.github/workflows‘ and added a file named ‘latex.yml‘.

2. Wrote the following workflow code:

```
name: Build and Release LaTeX Document

on:
  push:
    tags:
      - "*"

permissions:
  contents: write

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Repository
        uses: actions/checkout@v3

      - name: Set up LaTeX
        run: sudo apt-get update && sudo apt-get install -y texlive-full

      - name: Compile LaTeX Document
        run: pdflatex -interaction=nonstopmode -halt-on-error -file-line-error

      - name: Create Release
        id: create_release
        uses: actions/create-release@v1
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
        with:
          tag_name: ${{ github.ref_name }}
          release_name: Release ${{ github.ref_name }}
          draft: false
          prerelease: false

      - name: Upload Compiled PDF
        uses: actions/upload-release-asset@v1
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
        with:
          upload_url: ${{ steps.create_release.outputs.upload_url }}
          asset_path: ./assignment.pdf
          asset_name: assignment.pdf
          asset_content_type: application/pdf
```

3. Committed the workflow file and pushed it to GitHub.
4. Created a tag using the following commands to test the workflow:

   ```
   git tag v1.0
   git push origin v1.0
   ```

5. Verified that the workflow successfully compiled the LaTeX document and uploaded the PDF to the ”Releases” section of the repository.

2 Exploration Tasks

2.1 Vim Advanced Features

2.1.1 Macros

Macros allow you to record and replay sequences of commands, which is useful for repetitive tasks.

1. Press `q` followed by a register name (e.g., `q a` to record in register `a`).
2. Perform the actions you want to record.
3. Press `q` again to stop recording.
4. Replay the macro with `@a`.
5. Repeat multiple times using `N@a` (replace `N` with a number).

2.1.2 Visual Block Mode

Visual Block Mode allows you to select and manipulate text in a rectangular block.

1. Press `Ctrl + v` to enter visual block mode.
2. Use arrow keys to select a rectangular area.
3. Press `I` (insert) or `A` (append) to modify the block.
4. Hit `Esc` to apply the changes.

2.1.3 Splits and Buffers

Splits and Buffers allow you to work with multiple files or views simultaneously.

1. Open a horizontal split with `:split` or a vertical split with `:vsplit`.
2. Navigate between splits with `Ctrl + w` and arrow keys.
3. List open buffers with `:ls` and switch with `:b<N>` (replace `<N>` with the buffer number).

2.2 Memory Profiling

2.2.1 Memory Leak

A memory leak occurs when a program allocates memory dynamically but fails to release it after use. Over time, this can cause the system to run out of memory, leading to crashes or poor performance.

```
int *ptr = malloc(sizeof(int) * 100);
// Forget to free the memory: free(ptr);
```

2.2.2 Memory Profilers

Purpose of Valgrind: Valgrind is a tool that helps detect memory-related issues such as leaks, invalid accesses, and uninitialized memory.

- Identifies locations of memory leaks in your code.
- Tracks memory usage and helps optimize memory allocation.

Example Command:

```
valgrind --leak-check=full ./your_program
```

2.3 GNU/Linux Bash Scripting

2.3.1 fzf

What is Fuzzy Searching? Fuzzy searching matches approximate strings instead of exact matches. It's useful when you don't remember the exact name or details of what you're looking for.

Command Explanation:

```
ls | fzf
```

This lists all files and directories (`ls`) and pipes them to `fzf`, which allows you to interactively search and filter the results.

2.3.2 Using fzf to Find Your Favorite PDF

1. **Command to List All PDFs:**

   ```
   fd . --extension pdf
   ```

2. **Command to Use fzf for Selection:**

   ```
   fd . --extension pdf | fzf
   ```

2.3.3 Opening the File Using Zathura

Command to Open PDF:

```
zathura $(fd . --extension pdf | fzf)
```

This command combines `fd`, `fzf`, and `zathura`:

1. `fd` lists all PDFs.
2. `fzf` lets you interactively select one.
3. `zathura` opens the selected file.

3 Git and FOSS

3.1 README.md

3.2 Issues

3.3 FOSS Contribution