Computer Workshop
Final Assignment

Dr. Maleki Majd

Contents

1 1 Git and GitHub

1.1 Repository Initialization and Commits

I initialized the Git repository for this assignment using the following commands:

```
$ git init
$ git add .
$ git commit -m "Initialized repository for the final assignment."
```

1.2 GitHub Actions for LaTeX Compilation

Create a new file named `.github/workflows/main.yml` in the root of your repository. This file will define the GitHub Actions workflow. Open `main.yml` and add the following content:

```
name: Compile LaTeX

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - name: Set up Git repository
      uses: actions/checkout@v2

    - name: Compile LaTeX
      run: pdflatex -interaction=nonstopmode -halt-on-error main.tex
```

2 Exploration Tasks

2.1 Vim Advanced Features

1. **Multiple Cursors:** Vim supports multiple cursors, allowing you to edit multiple occurrences of a word simultaneously. To use this feature:

    - Place the cursor on a word.
    - Press `Ctrl` + `D` to select the word under the cursor.
    - Press `Ctrl` + `D` again to select the next occurrence.
    - You can now edit all selected occurrences simultaneously.

2. **Window Resizing:** Vim allows you to resize split windows dynamically. To resize a split window:

   - Press `Ctrl` + `W` and release.
   - Press `>` to increase width or `<` to decrease width.
   - Press `+` to increase height or `-` to decrease height.

3. **Vim Registers:** Registers in Vim allow you to yank and paste text selectively. Some advanced register usage includes:

   - `"ayy`: Yank the current line into register `a`.
   - `"ap`: Paste the contents of register `a`.
   - `":p`: Paste from the system clipboard (requires Vim with clipboard support).

2.2 Memory profiling

1. **Memory Leak:**

   In a program, a memory leak occurs when the program allocates memory dynamically (using functions like `malloc()` or `new`) but fails to release or deallocate that memory properly before the program terminates. This leads to a gradual accumulation of unused memory, and if the program runs for an extended period, it can eventually consume a significant amount of system memory. Memory leaks are a common source of performance issues and can cause programs to become slower or even crash.

   Memory leaks can happen due to various reasons, such as:

   - Forgetting to free allocated memory using `free()` or `delete`.
   - Losing track of pointers that point to allocated memory.
   - Failing to release memory in error or exceptional conditions.

   Detecting and fixing memory leaks is crucial for writing robust and efficient software.

2. **Memory Profilers:**

   Memory profilers are tools designed to help developers identify and analyze memory-related issues in their programs. One such powerful tool is **Valgrind**.

   Valgrind is a programming tool suite that provides various tools for memory profiling, including a memory leak detector. Its purpose is to assist developers in finding memory leaks, memory corruption, and other memory-related errors in C and C++ programs.

   How Valgrind helps when memory leaks happen:

- **Memory Leak Detection:** Valgrind can detect memory leaks by tracking memory allocations and deallocations during program execution. It reports any memory that was allocated but not freed.
- **Memory Error Detection:** Valgrind can identify memory-related errors such as accessing uninitialized memory, using memory after it has been freed, and other memory corruption issues.
- **Call Graphs and Profiling:** Valgrind provides detailed information about the program's memory usage, allowing developers to analyze memory behavior and optimize their code.
- **Dynamic Analysis:** Valgrind uses dynamic binary instrumentation to intercept memory-related function calls, providing a comprehensive view of memory usage during runtime.

By using Valgrind, developers can gain insights into the memory behavior of their programs, identify potential issues, and ensure efficient memory management.

2.3 GNU/Linux Bash Scripting

1. **2.3.1 fzf:**

   - **Fuzzy Searching:** Fuzzy searching is a technique that matches approximate patterns rather than exact ones. It allows finding results that are similar but not identical to the search query.
   - **Install fzf:** The command `ls | fzf` lists the files in the current directory and uses fzf for interactive, fuzzy selection. The selected file is then displayed on the terminal.

2. **2.3.2 Using fzf to find your favorite PDF:**

   (a) **List PDF files:** The command to list all files with the extension .PDF in the current directory is:

   ```
   find . -name "*.PDF"
   ```

   (b) **Select PDF with fzf:** The command to use fzf for selecting a PDF from the list is:

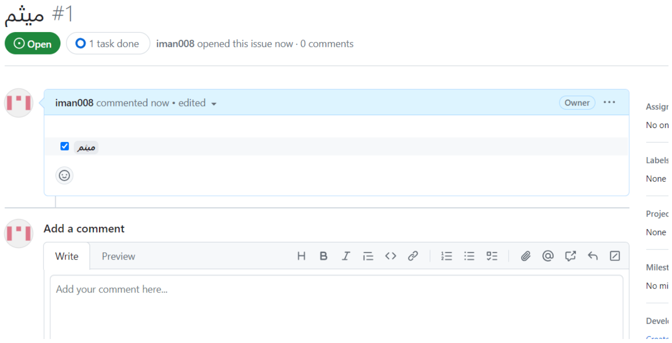
   ```
   find . -name "*.PDF" | fzf
   ```

3. **2.3.3 Opening the file using Zathura:**

   To open the selected PDF using Zathura, the command would be:

   ```
   zathura $(find . -name "*.PDF" | fzf)
   ```

3 Git and FOSS



ميثم #1

iman008 opened this issue now · 0 comments

iman008 commented now • edited

☑ ميثم

Add your comment here...

the link to github repo: `https://github.com/iman008/ComputerLabFinalAssignment`.