Web Server Development: Challenges and Solutions

1. Overview

This document outlines the errors and challenges encountered while developing a C-based web server on Windows and the solutions implemented to resolve them.

---

2. Compilation Errors and Solutions

2.1. `unrecognised emulation mode: i386pep` Error

Error Message:

```
C:\MinGW\bin/ld.exe: unrecognised emulation mode: i386pep
Supported emulations: i386pe
collect2.exe: error: ld returned 1 exit status
mingw32-make[1]: *** [Makefile:12: server.exe] Error 1
```

Cause:

The GCC toolchain was using an incompatible `ld.exe` (linker) version.

Solution:

- Ensured that `gcc` and `ld` versions were compatible.
- Used the correct MinGW-w64 toolchain (`x86_64-w64-mingw32`).
- Removed the `-m64` flag in the `Makefile` to match the correct architecture.

2.2. `fatal error: cannot find 'ld'`

Error Message:

```
collect2.exe: fatal error: cannot find 'ld'
compilation terminated.
mingw32-make: *** [Makefile:12: server.exe] Error 1
```

Cause:

The `ld` (linker) was missing from the system's PATH or not installed correctly.

Solution:

1. Checked if `ld.exe` existed by running: `where ld`
2. Found `ld.bfd.exe` instead of `ld.exe`.
3. Renamed `ld.bfd.exe` to `ld.exe` in `C:\msys64\mingw64\bin\`.
4. Updated the Makefile to use `ld.bfd` instead of `ld` (if renaming failed).
5. Installed missing linker binaries using:

   ```
   pacman -S mingw-w64-x86_64-binutils
   ```

6. Restarted the terminal and retried compilation.

---

3. Path and Environment Issues

3.1. Multiple GCC Versions in PATH

Problem:

Running `where gcc` returned multiple paths:

```
C:\msys64\mingw64\bin\gcc.exe
C:\MinGW\bin\gcc.exe
C:\gcc-14.2.0-no-debug\bin\gcc.exe
```

Solution:

- Ensured that `C:\msys64\mingw64\bin\` was prioritized in the PATH.
- Removed older or conflicting versions from the PATH.
- Verified the active GCC version using:

  ```
  gcc -dumpmachine
  ```

- Restarted the terminal to apply changes.

---

4. Makefile Issues

Error: Makefile Not Cleaning Properly

Issue:

```
mingw32-make clean
cmd /c del *.o lib\*.o server.exe
```

Cause: The clean command was not removing all object files properly.

Solution:

- Modified the clean rule in the `Makefile` to ensure all `.o` files and `server.exe` were deleted properly:
- ```
  clean:
      del /Q *.o lib*.o server.exe
  ```

- Verified the cleanup process by running `dir` after execution.

---

5. Lessons Learned

1. **Toolchain Consistency** – Ensuring that `gcc`, `ld`, and other components belong to the same toolchain prevents compatibility issues.
2. **Environment Variables** – Setting the correct PATH is crucial when working with multiple GCC installations.
3. **Dependency Management** – Using `pacman` to install missing packages (`mingw-w64-x86_64-binutils`) helped resolve linker issues.

General Debugging Steps Followed

- Used `where` command to locate missing executables.
- Verified system `PATH` to ensure correct compiler and linker usage.
- Checked MinGW installations and reinstalled necessary packages.
- Modified the `Makefile` to adapt to available tools.
- Used `pacman` to install missing dependencies in MSYS2.
- Restarted the system after each major change to apply updates.

---

6. Conclusion

This document summarizes the challenges encountered while compiling and running the web server and the solutions taken. By managing toolchain versions, resolving PATH conflicts, and correctly configuring the Makefile, I successfully set up a functional web server on Windows.
  ```