

Department of Computer Engineering  
University of Peradeniya  
CO1020 : Computer Systems Programming

Assembly Project

July 13, 2025

## 1 Mini Command-Line Shell

- **Duration:** 2 Weeks
- **Language:** ARM32 Assembly
- **Type:** Group Project (2 members)
- **Goal:** Build a minimal interactive shell in ARM32 assembly language.

## 2 Project Summary



Civilization has collapsed after a global AI uprising. Networks are down. Power grids are gone. Digital infrastructure? Burnt to dust. While scavenging the ruins of a former research bunker, your group finds a working ARM32-based embedded terminal. It's old, rugged, and mysteriously untouched. But there's a problem — it doesn't run Python, JavaScript, or even C. It only boots to a blank shell running raw ARM32 assembly. You, the last remaining Computer Engineer from the pre-AI era, in your survival group, have one task: Build a working command-line interface (CLI) in ARM32 assembly. This shell will be your group's lifeline to interact with the device and access vital systems.

## 3 Project Tasks

### 3.1 Task 1: Base Shell Architecture

- Use a single `shell.s` file to complete the project.
- Create and configure `.data`, `.bss`, and `.text` sections.
- Define `main` label as the program entry point.
- Implement a basic shell loop that continuously prompts for user input. The command prompt for the CLI may be `shell>`.
- Allocate a buffer in memory to store user input.
- Use system calls to print to the screen and read user input.

### 3.2 Task 2: Command Handling Logic

- Implement string comparison logic for command matching.
- Use conditional branching for command matching.
- Break logic into reusable functions. Create distinct code paths for each command.
- Save and restore registers using the stack within each function. Ensure that `lr` and temporary registers are preserved appropriately.
- Maintain the shell loop after each command is processed.
- Required commands are shown below.

Command	Action
hello	Prints "Hello World!"
help	Lists all commands available in shell
exit	Terminates the shell
clear	Clears the shell screen

Table 1: Basic Commands

### 3.3 Task 3: Custom Commands

- Design and implement 2 additional custom commands of your own, beyond the basic commands.
- These commands must be recognized and routed just like the basic ones.
- These commands should demonstrate more than minimal complexity.
- The following are examples of potential custom commands:

Command	Action
hex	Print a Number in Hexadecimal (Input given in the command line itself.)
avg	Average of N Numbers. (Input given in the command line itself.)

Table 2: Examples for Custom Commands

## 4 Marking Criteria

Section	Marks (%)
Basic Shell Loop and Prompt	10
Implementation of Basic Commands	20
Implementation of Custom Commands	30
Proper Stack Usage and Register Saving	10
Conditional Logic and Branching	10
Code Readability & Documentation	20

Table 3: Marking Breakdown

## 5 Submission

- Complete **shell.s** source code. Don't forget to include a lot of comments in the code!
- A 5-minute (maximum) recorded presentation (screenshare + voice) named **GroupXX\_assemblyproject.mp4** explaining the overall structure of your assembly code and also a working demo of your shell.
- A PDF report named **GroupXX\_assemblyproject.pdf** which includes an introduction, details about the commands of the CLI, implementation details about the code, explanations on design decisions, and a section for AI tool usage. Also, make sure to include a section on individual contributions.
- Include all of the above and submit a zip file named, **GroupXX\_assemblyproject.zip**

### Important Note on AI Tools:

Submitting code generated entirely by AI tools (e.g., ChatGPT, GitHub Copilot, etc.) **without demonstrating proper understanding or design effort—will result in zero marks.** *We evaluate not only the correctness of your code but also your understanding and ability to design and implement a solution.* Please ensure that your submission reflects your own learning. **Make sure to mention how, when, and where it was used and how you verified the correctness of its outputs in the report.**

## 6 Deadline

- The deadline is **27th July 2025 24:00h(midnight).**

## 7 Inquiries

- Please contact Thamish, (**thamishw@eng.pdn.ac.lk**) regarding any inquiries about the project.