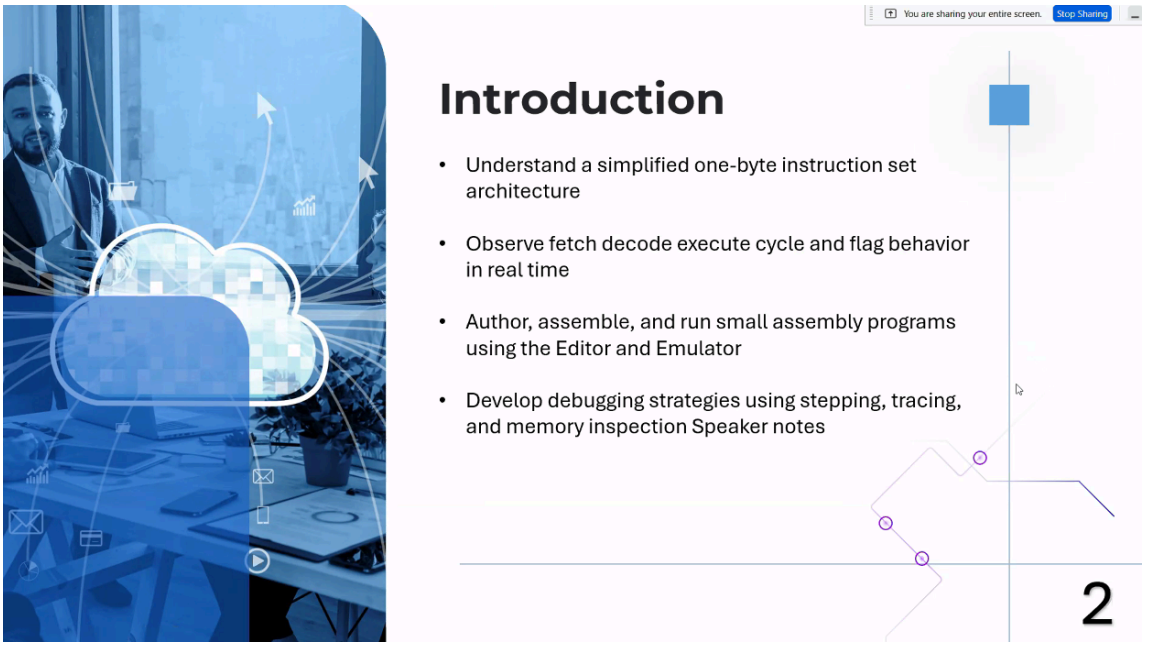
Status Ahmed 2025-10-29

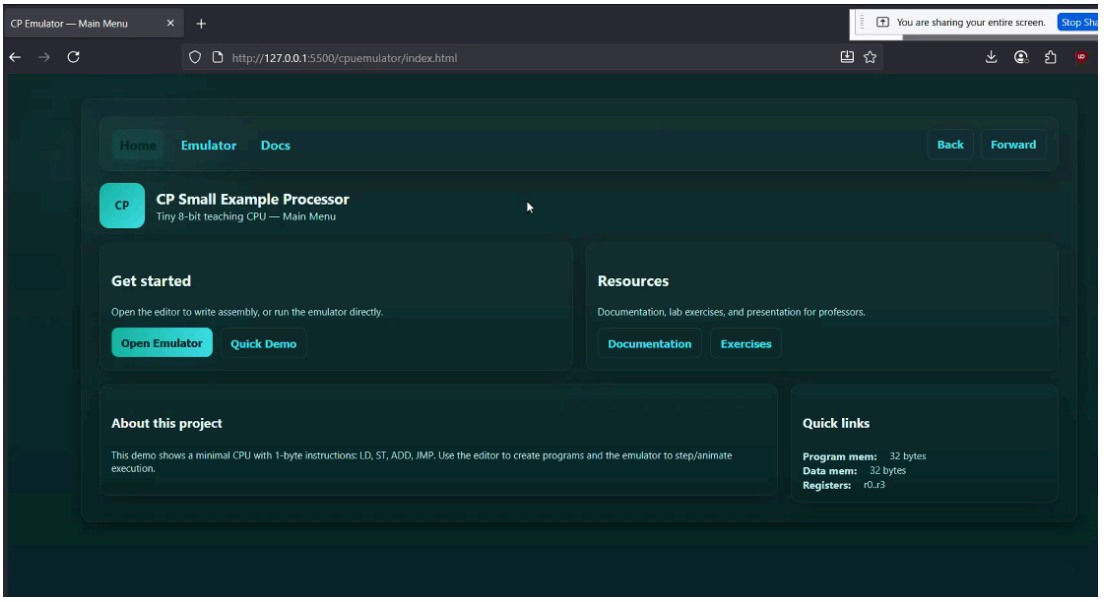
Agenda

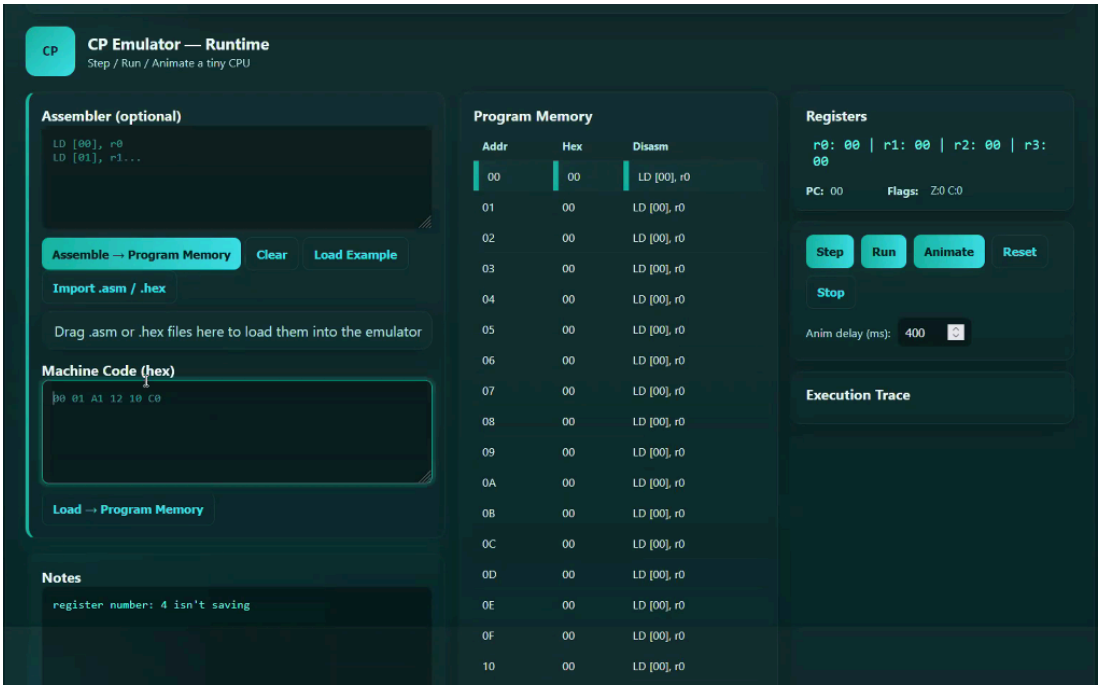
- As discussed last week:
- Ahmed will give a 10 min presentation on the current status of his work.
- We will discuss his presentation with him (5 min).
- Finally. we will define the next steps (teachers only, max. 15 min).

Presentation Ahmed

- 

  - Ahmed is new to bare metal, low-level code
  - this slide and the next just repeat the task 😞
- Demo

- 



- looks quite complete??

Proposed Editor Enhancements

Syntax-aware editor: syntax highlighting for mnemonics, registers, hex bytes; auto-indent; line numbers

Live diagnostics: inline parse errors, warnings for out-of-range addresses, and suggested fixes

UX improvements: undo/redo, search/replace, clipboard paste-to-assemble, keyboard shortcuts (Ctrl+S assemble)

- Ahmed wants to implement an editor with syntax completion ... - do we want that??
- Ahmed proposes to add group functions - do we want that??

Conclusion

Project summary: Built a compact browser-based toolchain consisting of an Editor and an Emulator that together let students write assembly, assemble to one-byte machine code, load .asm/.hex files, and observe execution state in real time.

Core learning outcomes achieved: students can assemble and run small programs, follow the fetch-decode-execute cycle, inspect registers and memory, and reason about flags and edge cases such as carry and wrap-around.

Key features delivered: syntax-agnostic assembler, program/data memory preview, step/run/animate controls, execution trace, drag-and-drop file import, notes with persistent save, and safe animation loop.

Discussion with Ahmed

- Stefan is happy!
- some remarks on slide 2 - simulation i not as precise as indicated
- please change memory layout to von Neumann (data and program memory in the same memory) - now it is Harvard (separate memories); maybe add a mode switch in the future
- please extend memory to 256 bytes; if disassembler outputs nonsense for unused memory adresses, it may do so😉
- Stefan likes demo!
- how to save memory? / how to save assembly? / .hex-file is nice, binary would be sufficient
- in the improved editor
  - no indentation needed
  - line numbers will be useful
  - highlight pc-address and operand address and used registers (source and destination in different colours)
- assembler is not correct!!
  - byte operands behind the opcode are missing
  - LD assembles but disassembly shows ST
- Will there be time to extend the instruction set?? - this is a lower priority, but the code should be written in a way that allows future extensions for different instruction sets
- With notes and program files and ... can we have something like a project that comprises everything to save and reload?
- currently we have basically a local javascript running in the browser; do we need server functionality?
  - Salah is happy, as long Stefan is happy😉
  - Ahmed has the idea to add server functionality, but so far there is no real need for server functions

next meeting 5.11. 16:00 - no Malek