ARM program

CSC 236

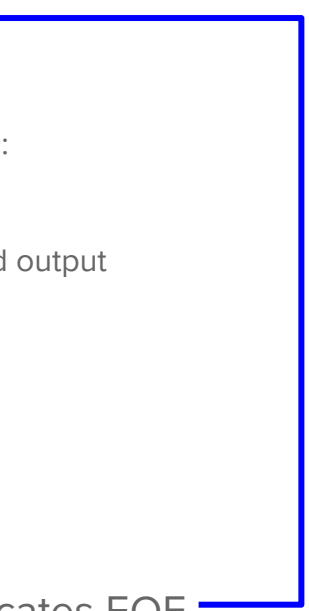
ARM program

- Read specification
  - Due date: see syllabus
- Team assignment (optionally)

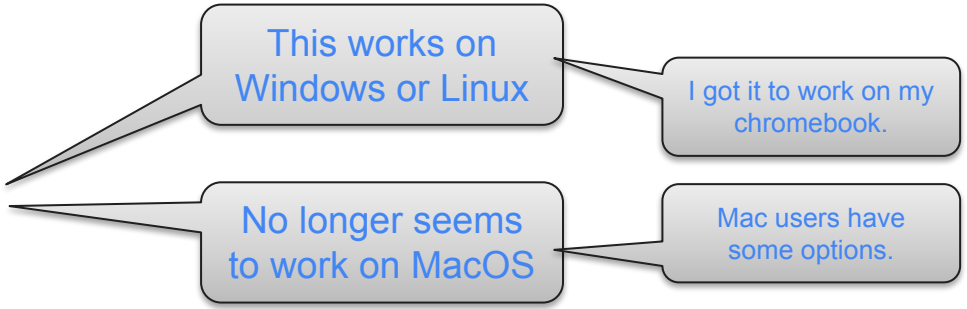
Assignment

- Write the “KEY” program in ARM
  - See KEY assignment for details
  - One difference
    - Ignore ‘.’ (period)
    - We can just read until the end-of-file

- Read a line of ASCII text
  - Input and output strings
  - For each character in input string :
    - uppercase — output
    - lowercase — capitalize and output
    - blank – output
    - other — ignore
  - At end of input string
    - write output string to file
    - write cRLF to file
- Repeat till read string SWI indicates EOF
- Close input & output file



Steps



- Step 0 — install ARMsim
- Step 1 — create design
  - Model after KEY (same code organization, different instructions)
  - Use hello.s, copystr.s, copyfile.s as examples
  - Think about conditionally executed instructions
- Step 2 — code solution
  - Name file armkey.s

Steps

- Step 3 — test/debug
  - Use sample file key.in
    - Modify for additional tests
  - Use ARMsim to read, assemble, run
  - Output data into key.out
  - Verify key.out is correct
- Step 4 — grade
  - Follow instructions in specification
  - Need to execute in ARMsim and grade in DOSbox
- Step 5 — submit
  - arm.ans — created by gradarm.exe (in DOSbox)

Design

- In 8086, maybe you used xlat
  - al = bx[al]; // C syntax
  - al gets the value at memory location bx+al
- ARM doesn't have that instruction
  - But, you can implement it yourself
  - ... using indirect addressing.
  - `ldr r1, [r2, r3]   ; r1 = [r2+r3]`
    - `Use one register for the start of the translation table`
    - `Another for the offset into the table.`