

# CMPS 455

October 12, 2021

## Project 3 - Running User Programs

From *Jason Woodworth's Tutorial Part 5*

### Running User programs

- `../userprog/..`
  - run user program
  - `../nachos -x "../test/halt"`
- `../test/matmult` ← matrix multiplication
- `../test/sort` ← good for testing effects of page swapping algo, etc. because it has a lot of ticks
- `../test/shell` ← runs a shell, allows you to run other programs..
- as of now, main memory is set to zero when user prog is run
- second program will overwrite previous programs main memory
  
- `main.cc`
  - `Initialize()` ← `system.cc`
  - `#IFDEF userprog`
    - `StartProcess()` - start process with filename given as param
      - brings you to `../userprog/progtest.cc`
    - `AddrSpace *space`
      - define page tables, reading code from user program
    - `space = new AddrSpace(executable)`
      - and `machine→Run()` are most important part of `StartProcess()`
    - `currentThread→space = space`
    - `machine→Run()`
- `system.cc`
  - second `#IFDEF userprog`
    - `"-s" arg`
    - EX: `../nachos -x "../test/halt" -s`
      - this will debug, giving you a cycle by cycle program counter number
      - state of registers (`PCReg`, `NextPCReg`, `PrevPCReg`)
  - third `#IFDEF userprog`
    - handles machine directory
    - enables machine
    - carries all stuff for MIPS
    - main memory array

- instructions for machine code, etc.
  - `addrspace.cc`
    - "where the real work begins"
    - Noff header is defined
    - `executable→ReadAt()`
      - reads from executable into first param
      - reads `sizeof(noffH)` amount, followed by offset (2nd and 3rd params)
      - "probably need to explore this more in `filesys`"
        - I assume he means for later tasks
      - `//How big is address space?`
        - `size = .....`
          - how much room in main memory we will be taking up
            - sort of at least, by prog instructions
          - `noffH.initData.size`
            - EX: `int i = 10;`
          - `noffH.uninitData.size`
            - EX: `int i;`
          - `UserStackSize` is constant, dont worry about it
        - need to change `ASSERT` below here
        - `pageTable = new TranslationEntry()`
          - a class that translates virtual page to phys page
          - physical page def needs to be changed
        - `executable→ReadAt()`
          - change this to not just start at 0 (beginning of `mainMemory`)
            - the page off set
          - 80% of changes go in `lines 106-119`
- open executable → load in main mem → run MIPS by telling it where PC counters and data is in `mainMemory`