Midterm I: May 3rd

Lessons from HW1?

- Windows does a lot in the background (updater, virus scanner, etc.). Under load not so.
- A lot of disc accesses and cache misses while idle.
- Page faults happen in clusters. A page fault on one process then it happens on others at the same time.
- There are a lot of context switches that occur.
- Not a lot of threaded parallelism
- Most CPU usage from the idle process -> accounting only
- Windows telemetry uses your CPU, network and your data
- Windows Defender caused the disc usage to go up to 100% and stay there
- Background "bloatware" took more CPU than the foreground process in use
- Chrome was eating a ton of CPU time. Launched 10 different processes to run a single video
- 4K streaming consumed far more CPU time than 1080 video

RISC v CISC
Why did CISC happen?

• Marketing: More instructions, more better
• Code density
• Memory was expensive & slow (and still is)
• Support HLL (High Level Language) *debatable!
• Expose cool tricks to the user

What & Why is RISC?

• Fixed length instructions – simplified decoding logic and instruction encoding
• Small number of instructions – less effort for the compiler, speed (faster), less hardware to support
• More easily utilize ILP
  • e.g. ADD BX, DX; AX; [BX + DI] = [BX + DI] + AX; LOAD AX, [BX + DX]; ADD AX, [BX + DX], AX
  • STORE [BX + DX], IX
• Simpler instructions are easier to pipeline
• By not supporting a lot of instructions can make better use of silicon area elsewhere (definitely true in 80s and 90s)
• Not microprogrammed (there is a “PAL” code)
• Easier to design
• Less prone to errors
• General purpose registers