

2/13/2023

test prime-ss - different prime test from assignment

- random keys
- generates #64 random ^{-salim strawman}
- we've done miller-Rabin *
- we'll use gnu multiprecision to generate random
- write utility functs
 - is even, is odd
 - power mod
 - Jacobi symbol - don't have to do
 - is prime
 - probabilistic primality test
 - run multiple times to increase accuracy

"when you look @ number theory, you see God's shadow."

- pick random # & check if prime very is prime
- density of primes is approx logarithmic

- Carmichael pseudo prime
- rare

- greatest common divisor
 - $\Theta(n) = \log n$
 - euclidean algorithm
- least common multiple
- inverse (multiplicative) $a \times b = -1$
 - use of cd
 - mod n #s

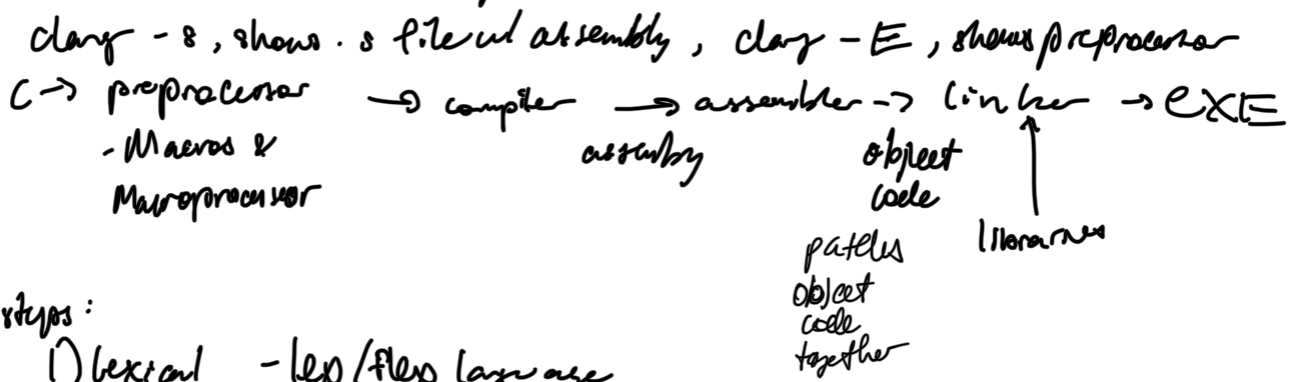
- write some of utility

- write it w/ 64 bits, then replace w/ gnu multiprecision
- in python too

Compilers:

MS A1 8080 - press switch for address, then data.

- old computer



Compiler steps:

- 1) Lexical - lex/flex language
- 2) Syntax - changes int, {, (, and other symbols
 - uses parse trees, to analyse syntax
 - abstract syntax tree
 - recursively traverse tree

3) Translator

- translation to machine code from assembly
- creates tokens

4) Optimisation

5) Storage assignment

6) Code generation

7) Assembly phase

Loader - lives in operating system

- allocates memory
- linker - resolves symbolic references
- relocation - changes necessary dependencies
- loading place machine code & data into process

man.exe

python, java script - interpreted languages

- translates line by line

interpreter - walks through syntax tree & just does it

interpreted languages are slower

bash is interpreted language