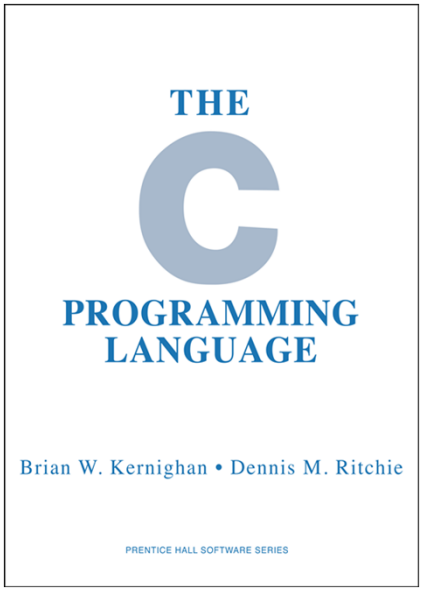
Lidt C historik

- Udviklet i perioden 1969 - 1973 som højniveau sprog til programmering af Unix af (mest) Dennis Ritchie
- Udviklet da computere var store, dyre
- Original maskine (DEC PDP-11) havde lille kapacitet
  - 24K byte arbejds-lager, 12K brugt til OS
  - Meget lidt plads til Compiler og OS
  - Robusthed, sikkerhed mindre vigtigt
- Designet til system- or maskinnær programmering
  - Operativ systemer, især den centrale ”kerne”
  - Tilhørende værktøjs/kommando-linie programmer
  - Compilere
- Udviklet fra B, som er udviklet fra BCPL/Algol60

C sproget - 1

- Alt-dominerende sprog inden for system- og maskinnær programmering
  - Operativ systemer, firmware, indlejrede systemer, IoT enheder
  - Udbredt: C compilere findes til alle processorer!
  - C er kongen!
- Lav-niveau programmering - “Højniveau assembler”?
  - Gennemskuelig oversættelse til maskin-kode
  - Nemt at interface til assembler
  - Direkte manipulation af adresser, hukommelses-indhold, bits og bytes
  - Program kan laves portabelt (ved korrekt anvendelse)
- Compiler producerer effektiv kode (kompakt og hurtigt)
- Oversættelse kan ske i et enkelt gennemløb af kilde-teksten: “Single-pass”
  - Alle symboler skal erklæres før brug (som compileren læser programmet).
- Standardisering og varianter:
  - I 1978 skriver Dennis Ritchie and Brian Kernighan “C-bogen”
    - The C Programming Language, Prentice-Hall. ISBN 0-13-110163-3.
  - I 1982 American National Standards for Information Systems (ANSI) etablerer komité for at lave en C-standard. Resulterer i C89.
  - Ratificeret i 1990 af ISO (C90)
  - C standarden er revideret gentagende gange: 1999 (C99), 2011 (C11), ‘18 (C18), og 202x (C2x)



C sproget - 2

- "C is quirky, flawed, and a tremendous success", [Dennis Richie]
  - Programmer kan skrives så der er fuldstændigt ulæselige
  - Forkert anvendelse giver mange kilder til fejl og sikkerhedshuller
  - Dårligt specificeret
    - Mange programmer med korrekt syntax har udefineret adfærd
    - Mange "finurligheder" som er udefinerede
    - => uforudsigeligt resultat

C sproget - 3

- *”A **low-level programming language** is a programming language that provides little or no abstraction from a computer's instruction set architecture—commands or functions in the language map closely to processor instructions.”* [Wikipedia]
- Men mange aspekter af moderne maskiner, som er vigtige for programmøren, er der ikke abstraktioner for i C !!!!!!!
  - Caches i flere niveauer
  - Instruktions-niveau parallelitet, spekulativ udførelse
  - Parallelitet, og fler-kerner
  - Virtuel hukommelse
  - En optimerende C compiler er kompliceret og genererer ikke ”nemt gennemskuelig” assembler
- Andre nyere system-niveau sprog
  - D https://dlang.org/overview.html
  - Rust https://www.rust-lang.org/
  - Kommende udbredelse ??????

C sproget i CA-OS

- Illustrere hvordan programmer oversættes og afvikles på maskinen
- Illustrere hvordan maskinens virkemåde smitter af på hvordan programmet (bør) skrives
- Eksemplificere ”parallel programmering”

- Assembler er tit for tungt – kan spille sammen med C-programmer
- Vi skal ikke skrive store programmer

- Men det er nødvendigt at kunne læse og skrive små C-program fragmenter
  - …og arbejde med C oversættelsesværktøjer som GCC, GDB, ASM, …

Dele af C som der er vigtigt I kan

- Simple data-typer og konvertering: char, Int, long, float, double
- Sammensatte typer: arrays (1d og 2d), structs
- Pointer begrebet
- Kontrol strukturer iteration: for, while, do-while, goto
- Kontrol strukturer for selektion:
  - if-then-else,
  - betingede udtryk: (Exp)?Exp_1:Exp_2
- Udtryk og aritmetik: logiske betingelser og tildeling
- Funktioner med parametre, lokale og globale variable