Programación en C

Introducción

Me

Tomás Perez Molina

Universidad Austral - Politecnico di Milano

Multiprocessing - Cybersecurity - Compiladores - AI

Tesis: Extensiones a MANGO

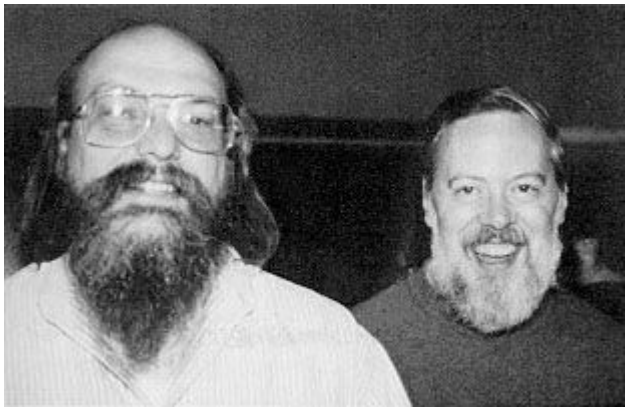
tomas.perezmolina@ing.austral.edu.ar

https://github.com/Tomas-Perez

http://www.mango-project.eu/

¿Qué es C?

- Lenguaje de programación imperativo
- De propósito general
  - Systems programming
- De alto nivel, pero cercano al hardware
  - “Mid level”
- Creado en 1972 en Bell Labs por Thompson & Ritchie.
- En 1973 C es el 1er lenguaje de alto nivel usado para programar un sistema operativo, Unix v4.



¿Qué es C?

- En 2020:
  - #1 en el index TIOBE
    - Índice de popularidad del lenguaje en internet determinado por la cantidad de búsquedas.
    - https://www.tiobe.com/tiobe-index/
  - #8 en github
    - Popularidad de proyectos open source
    - https://octoverse.github.com/
- Es el núcleo de todos los SO más usados (Windows, Linux, MacOS)

¿Por qué usar C (hoy)?

- Performance
- Portabilidad
- Interoperabilidad
- Drivers
- Bare metal (Sin SO)
- Sistemas operativos
- Sistemas embebidos

¿Por qué NO usar C?

- Falta de abstracciones
- Control de detalles de bajo nivel
- Seguridad
- Undefined behavior

El estándar

La popularidad del lenguaje en sus comienzos introdujo muchas versiones custom que eran incompatibles entre sí.

Esto requirió la estandarización del lenguaje para garantizar compatibilidad entre compiladores.

Al día de hoy existen 4: ANSI C, **C99**, C11 y C17.

Undefined behavior

A diferencia de otros lenguajes, el estándar de C no define qué debería ocurrir en cualquier posible situación.

Dependiendo la arquitectura podría ser problemático para performance o incluso imposible de implementar.

Esto introduce:

1. Unspecified behavior: 2 o más alternativas
2. Implementation-defined behavior: depende del compilador
3. Undefined behavior: ninguna garantía

¿Por qué aprender C?

Para poder programar en C es necesario tener conocimiento de cómo funciona el lenguaje detrás de escenas.

Lo que incluye:

- Interacción con el sistema operativo
- Interacción con el hardware
- Compilación

Estos temas pueden ser aplicados a cualquier lenguaje.

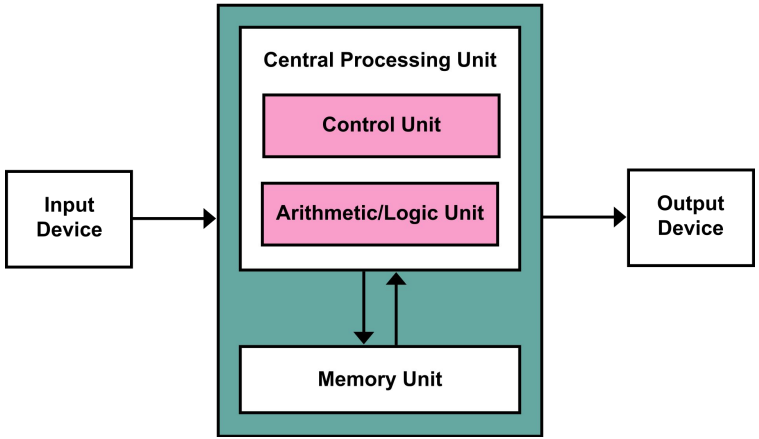
Arquitectura Von Neumann

Modelo abstracto de la arquitectura de una computadora.

- CPU (control, aritmética y lógica)
- Memoria
- Input/Output

Ya estamos familiarizados con el funcionamiento del CPU.

Nos falta cubrir los otros dos.



Plan de curso

Vamos a estudiar como funciona C para poder utilizar las herramientas del lenguaje de la manera correcta y aprovecharlas al máximo.

Los temas que vamos a cubrir son:

- Manipulación de memoria
- Alocación de memoria
- Input/Output

Ejemplos y consultas

Todos los ejemplos que vamos a ver en clase van a estar en el siguiente repositorio:

https://github.com/Tomas-Perez/seminar

Para consultas sobre el curso pueden abrir un nuevo issue.

- Todos tienen acceso a las preguntas y respuestas
- Filtros, búsqueda
- Code highlighting, imágenes, links, etc

https://guides.github.com/features/mastering-markdown/

Trabajo final

A definir.

Probablemente involucre analizar y arreglar un programa en C. La entrega incluiría el código arreglado más un pdf explicando los problemas presentes y cómo se resolvieron.

Temas como:

- Algoritmos
- Diseño

Son cubiertos por otras materias.