

ÍNDICE

Índice	1
I Clase 1: Introducción/Motivación	2
1. Presentación	2
2. Propósito del curso	2
3. Resultados de aprendizaje	2
4. Acerca del profesor	2
5. ¿Por qué aprender a programar en C?	3
6. ¿Por qué aprender assembler Risc-V?	3
7. ¿Por qué aprender un sistema operativo basado en Linux?	3
8. ¿Qué es la programación de software de sistemas?	4
9. Programa del curso	4
10.Bibliografía	4
11.Evaluación	4
12.Consejos y reglas importantes	5
13.Trabajo para la casa	5
Referencias	6



CLASE 1: INTRODUCCIÓN/MOTIVACIÓN

1 Presentación

Estos son los apuntes no oficiales del curso CC3301-1 cursado en Primavera 2022 bajo las cátedras del docente Luis Mateu B. El material usado es directamente obtenido de cátedras, en adición a los materiales del curso. [1].

2 Propósito del curso

El propósito del curso CC3301 Programación de software de sistemas (PSS) es que los/las estudiantes escriban y mejoren programas en lenguaje C, que requieren hacer un uso eficiente de la plataforma, y por lo tanto necesitan conocer su arquitectura de hardware y la interfaz de programación de aplicaciones (API) del sistema operativo.

3 Resultados de aprendizaje

En este curso aprenderán a:

- Escribir programas eficientes en el lenguaje C
- Usar el *debugger* **ddd** para diagnosticar errores
- Usar *sanitize* para detectar errores de programación como fugas de memoria o punteros locos
- Identificar a qué instrucciones assembler **Risc-V** se compilan los programas en C
- Escribir trozos de programas en assembler **Risc-V**
- Escribir programas que usan directamente los servicios provistos por *Linux* (el núcleo de sistemas operativos como *Debian*, *Ubuntu*, *Android*, etc.)
- Crear procesos paralelos para aprovechar todos los cores del computador y así disminuir el tiempo de ejecución

4 Acerca del profesor

Resumen bajo palabras de Mateu.

- Profesor de “jornada parcial” del DCC
- Solo me dedico a los 3 cursos que dicto para la facultad
- Los otros 2 cursos son arquitectura de computadores y sistemas operativos
- Trabajé 5 años en Synopsys: empresa con sede en California y centros de investigación y desarrollo en varios países, incluyendo uno en Chile
- Synopsys es líder en software para diseñar circuitos digitales, usado por Intel, AMD, nVidia, Qualcomm, etc.
- Trabajé en el mantenimiento de Design Compiler, un compilador de Verilog/Vhdl de unas 10 millones de líneas de código en C que corre principalmente bajo Linux

5 ¿Por qué aprender a programar en C?

- Es el lenguaje preferido cuando se necesita eficiencia, por ejemplo para programar un decodificador de video
- Existe una amplia base de software escrito en C y que necesita mejorarse:
 - ✓ Resolver bugs
 - ✓ Agregar funcionalidades
 - ✓ Portar a nuevas plataformas
 - ✓ Mejorar la eficiencia
- Ejemplos: los núcleos de Linux (C) y Windows (C y C++), el intérprete de Python, el compilador just-in-time de Java, administradores de bases de datos, etc.
- Desventajas:
 - × No es robusto
 - × Es inseguro
- Futuro ficción: será reemplazado por Rust

6 ¿Por qué aprender assembler Risc-V?

- Porque es el assembler más fácil de aprender
- Hace más fácil aprender assemblers más complejos como x86 y Arm
- Se requiere programar en assembler para usar de la manera más eficiente las instrucciones especializadas de los procesadores como manejo de vectores
- Algunos bugs son tan complejos que requieren revisar el assembler generado por el compilador
- Risc-V se posiciona en términos de uso como la tercera plataforma después de Arm y x86
- Y no parará de crecer porque un fabricante de chips no necesita pagar una licencia por usar el set de instrucciones de Risc-V
- Hay diseños avanzados de Risc-V open source y gratuitos
- Diseñado para que los fabricantes puedan agregar instrucciones aceleradoras del cálculo de operaciones gráficas, de redes neuronales, de encriptación, etc.

7 ¿Por qué aprender un sistema operativo basado en Linux?

- Linux es una reimplementación del núcleo del sistema operativo del legado Unix
- Android, el sistema operativo de los celulares, usa un núcleo derivado de Linux
- iOS y OS X (de Apple) son derivados de Unix
- La mayoría de los servidores en Internet usan un sistema operativo basado en Linux: CentOS, RedHat o Debian
- Los notebooks y computadores de escritorio típicamente usan el sistema operativo Windows, pero WSL 2 de Microsoft también permite correr en Windows los sistemas operativos Debian, Ubuntu y otros basados en Linux, con todas sus aplicaciones

8 ¿Qué es la programación de software de sistemas?

De acuerdo a [Wikipedia](#) se trata de programar software:

- Que provee servicios a otro software
- Que requiere alto desempeño
- O ambos

Ejemplos: núcleos de sistemas operativos, compiladores, aplicaciones de ciencia computacional, motores de juegos, automatización industrial, etc.

Típicamente se programan en el lenguaje C

Lo opuesto es la programación de software de aplicaciones como un sistema de administración para una empresa, scripts para páginas web, un juego de salón, etc.

Finalmente: ¡es subjetivo!

9 Programa del curso

- Programación en el lenguaje C: tipos, sintaxis de las instrucciones, direcciones de memoria, punteros, errores típicos
- Arquitectura de computadores: Risc vs. Cisc, assembler Risc-V, compilación de programas en C a instrucciones Risc-V
- Implementación de la CPU: circuitos digitales, la memoria, diseño de la cpu, ejecución secuencial, en pipeline, superescalar y fuera de orden.
- Linux: estudio de la API de Linux/Unix (application programming interface), el shell de comandos, cómo se crean los procesos y se cargan los programas, paralelización con múltiples procesos

10 Bibliografía

- Se publicarán en la sección novedades de U-cursos:
 - ✓ Videos de las clases de este semestre y/o de semestres anteriores
 - ✓ Pdf de las presentaciones
 - ✓ Material para probar los ejemplos de las clases
- En página Web: <https://users.dcc.uchile.cl/~lmateu/CC3301/>
 - ✓ Instrucciones para instalar la distribución oficial de Linux en este curso: Debian 11
 - ✓ Controles de semestres pasados
- Material complementario:
 - ✓ [Apuntes del curso](#) (actualización necesaria)
 - ✓ Kernighan, B y Ritchie, D (1988) “[The C Programming Language](#)”
 - ✓ Richard Stones, Neil Matthew (2003) “[Beginning Linux Programming \(Programmer to Programmer\)](#)”

11 Evaluación

En página Web: https://www.u-cursos.cl/ingenieria/2022/1/CC3301/1/datos_curso/

12 Consejos y reglas importantes

- No intente resolver las tareas sin haber estudiado la materia correspondiente primero. Cometerá errores que no será capaz de entender y terminará perdiendo más tiempo que el que se ahorró al no estudiar.
- Si sobrepasa el tiempo nominal publicado para resolver la tarea, pida ayuda.
- Pedir ayuda no es copiar cuando el código lo escribió Ud. mismo.
- Copiar un fragmento de código de la tarea de un compañero sí es copia.
- Puede pedir ayuda a un compañero.
- O al profesor de cátedra o los auxiliares en alguno de los horarios de consultas o por correo
- Una tarea en C puede correr exitosamente en su computador y fallar completamente en plataformas ligeramente distintas. Su tarea será corregida bajo Debian 11. Asegúrese de que funciona en esa plataforma.

13 Trabajo para la casa

- Instale el Linux oficial del curso (Debian 11): siga las instrucciones que aparecen en [esta página](#) en la sección cómo correr Debian en su computador.
- Estudie [este tutorial](#) que le enseñará los comandos básicos de Linux. Abra un terminal y experimente con estos comandos.
- Antes de la clase del jueves: estudie en los apuntes del curso la sección principios básicos del lenguaje C
 - Compile y ejecute todos los ejemplos dados
 - Complete el ejercicio final: factorial

REFERENCIAS

- [1] Mateu, Luis: *Programación de Software de Sistemas - Novedades*. <https://www.u-cursos.cl/ingenieria/2022/1/CC3301/1/novedades/>, 2022.