

Programación de Microcontroladores

Clase 1

Carrera de Especialización en Sistemas Embebidos - FIUBA

Esp. Ing. Patricio Bos

versión: 2017-02-8 rev 0.1



Organización de la presentación

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

1 Introducción

- Consideraciones del Curso

2 Repaso de C

- Motivación para usar C
- Consideraciones de performance

3 C para sistemas embebidos

- Control de flujo



Disclaimer

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- Las presentaciones de este curso están basadas en el material de la asignatura “Sistemas Digitales” de la UNQ a cargo del prof. Ing. José Juárez.
- Los ejemplos de código fueron recopilados y preparados por el Esp. Ing. Eric Pernia.



Disclaimer

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- Las presentaciones de este curso están basadas en el material de la asignatura “Sistemas Digitales” de la UNQ a cargo del prof. Ing. José Juárez.
- Los ejemplos de código fueron recopilados y preparados por el Esp. Ing. Eric Pernia.



Condiciones de aprobación

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- 6 ejercicios prácticos de entrega obligatoria. Plazo máximo de entrega 2 semanas desde la entrega del enunciado.
- Examen teórico escrito multiple-choice en la última clase.
- 75 % de asistencia a las clases.



Condiciones de aprobación

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- 6 ejercicios prácticos de entrega obligatoria. Plazo máximo de entrega 2 semanas desde la entrega del enunciado.
- Examen teórico escrito multiple-choice en la última clase.
- 75 % de asistencia a las clases.



Condiciones de aprobación

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- 6 ejercicios prácticos de entrega obligatoria. Plazo máximo de entrega 2 semanas desde la entrega del enunciado.
- Examen teórico escrito multiple-choice en la última clase.
- 75 % de asistencia a las clases.



Bibliografía

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- The C programming language K& R
- Expert C programming Deep C secrets
- The C puzzle book



Bibliografía

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- The C programming language K& R
- Expert C programming Deep C secrets
- The C puzzle book



Bibliografía

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- The C programming language K& R
- Expert C programming Deep C secrets
- The C puzzle book



Motivación para usar C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para sistemas embebidos

Control de flujo

- Lenguaje estructurado de “mediano” nivel (o assembler de alto nivel).
- Razonablemente fácil de aprender, comprender, programar y depurar.
- Acceso directo a las APIs de bajo nivel del HW (seguramente escritas en C).
- Es procesador-independiente y no está atado a ningún microprocesador en particular.
- Existencia de compiladores gratuitos para prácticamente todos los microprocesadores del mercado.
- Extenso soporte (bibliografía, cursos, artículos, foros, etc...).



Motivación para usar C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- Lenguaje estructurado de “mediano” nivel (o assembler de alto nivel).
- Razonablemente fácil de aprender, comprender, programar y depurar.
- Acceso directo a las APIs de bajo nivel del HW (seguramente escritas en C).
- Es procesador-independiente y no está atado a ningún microprocesador en particular.
- Existencia de compiladores gratuitos para prácticamente todos los microprocesadores del mercado.
- Extenso soporte (bibliografía, cursos, artículos, foros, etc...).



Motivación para usar C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- Lenguaje estructurado de “mediano” nivel (o assembler de alto nivel).
- Razonablemente fácil de aprender, comprender, programar y depurar.
- Acceso directo a las APIs de bajo nivel del HW (seguramente escritas en C).
- Es procesador-independiente y no está atado a ningún microprocesador en particular.
- Existencia de compiladores gratuitos para prácticamente todos los microprocesadores del mercado.
- Extenso soporte (bibliografía, cursos, artículos, foros, etc...).



Motivación para usar C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- Lenguaje estructurado de “mediano” nivel (o assembler de alto nivel).
- Razonablemente fácil de aprender, comprender, programar y depurar.
- Acceso directo a las APIs de bajo nivel del HW (seguramente escritas en C).
- Es procesador-independiente y no está atado a ningún microprocesador en particular.
- Existencia de compiladores gratuitos para prácticamente todos los microprocesadores del mercado.
- Extenso soporte (bibliografía, cursos, artículos, foros, etc...).



Motivación para usar C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- Lenguaje estructurado de “mediano” nivel (o assembler de alto nivel).
- Razonablemente fácil de aprender, comprender, programar y depurar.
- Acceso directo a las APIs de bajo nivel del HW (seguramente escritas en C).
- Es procesador-independiente y no está atado a ningún microprocesador en particular.
- Existencia de compiladores gratuitos para prácticamente todos los microprocesadores del mercado.
- Extenso soporte (bibliografía, cursos, artículos, foros, etc...).



Motivación para usar C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- Lenguaje estructurado de “mediano” nivel (o assembler de alto nivel).
- Razonablemente fácil de aprender, comprender, programar y depurar.
- Acceso directo a las APIs de bajo nivel del HW (seguramente escritas en C).
- Es procesador-independiente y no está atado a ningún microprocesador en particular.
- Existencia de compiladores gratuitos para prácticamente todos los microprocesadores del mercado.
- Extenso soporte (bibliografía, cursos, artículos, foros, etc...).



Eficiencia y desempeño del lenguaje C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

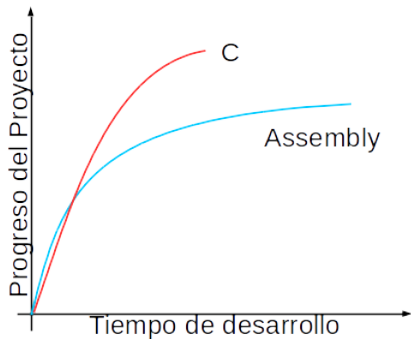
Consideraciones
de performance

C para

sistemas
embebidos

Control de flujo

- La eficiencia más alta en cuanto a tamaño de código y velocidad de ejecución se logra en assembler. Un compilador de C puede acercarse a esta eficiencia si está optimizado para la arquitectura del uC.
- Comparado con assembler, en C se puede generar código más confiable, portable, mantenible y escalable.
- Comparado con otros lenguajes de alto nivel, en C se puede generar código más compacto y eficiente.





Eficiencia y desempeño del lenguaje C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

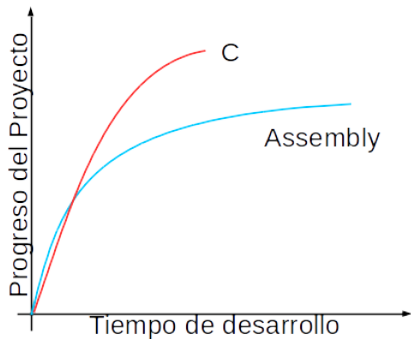
Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- La eficiencia más alta en cuanto a tamaño de código y velocidad de ejecución se logra en assembler. Un compilador de C puede acercarse a esta eficiencia si está optimizado para la arquitectura del uC.
- Comparado con assembler, en C se puede generar código más confiable, portable, mantenible y escalable.
- Comparado con otros lenguajes de alto nivel, en C se puede generar código más compacto y eficiente.





Eficiencia y desempeño del lenguaje C

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

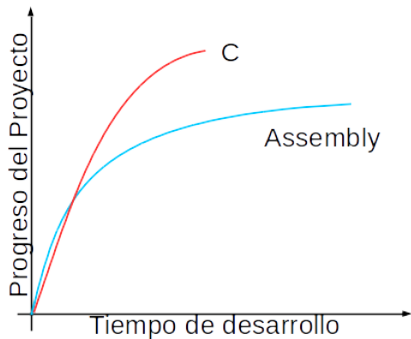
Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- La eficiencia más alta en cuanto a tamaño de código y velocidad de ejecución se logra en assembler. Un compilador de C puede acercarse a esta eficiencia si está optimizado para la arquitectura del uC.
- Comparado con assembler, en C se puede generar código más confiable, portable, mantenible y escalable.
- Comparado con otros lenguajes de alto nivel, en C se puede generar código más compacto y eficiente.





Ejemplos de optimización

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- uC con arquitectura basada en acumuladores trabajan mejor con variables globales.
- uC con arquitectura basada en registros trabajan mejor con variables locales.
- Las variables locales permiten mejor modularidad y por lo tanto código más portable y reutilizable.
- Si el desempeño de una sección de código es crítica se puede hacer en assembler.



Ejemplos de optimización

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- uC con arquitectura basada en acumuladores trabajan mejor con variables globales.
- uC con arquitectura basada en registros trabajan mejor con variables locales.
- Las variables locales permiten mejor modularidad y por lo tanto código más portable y reutilizable.
- Si el desempeño de una sección de código es crítica se puede hacer en assembler.



Ejemplos de optimización

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- uC con arquitectura basada en acumuladores trabajan mejor con variables globales.
- uC con arquitectura basada en registros trabajan mejor con variables locales.
- Las variables locales permiten mejor modularidad y por lo tanto código más portable y reutilizable.
- Si el desempeño de una sección de código es crítica se puede hacer en assembler.



Ejemplos de optimización

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

- uC con arquitectura basada en acumuladores trabajan mejor con variables globales.
- uC con arquitectura basada en registros trabajan mejor con variables locales.
- Las variables locales permiten mejor modularidad y por lo tanto código más portable y reutilizable.
- Si el desempeño de una sección de código es crítica se puede hacer en assembler.



Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para
sistemas
embebidos

Control de flujo

control de flujo directivas precompilador tipos prefijos sufijos modificadores
operadores aritméticos operadores de bit



Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance



C para sistemas embebidos

Control de flujo



Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance



C para sistemas embebidos



Control de flujo



Tipos de variables

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para sistemas embebidos

Control de flujo

- `global`: Almacenamiento global y visibilidad global.
- `static`: Almacenamiento global y visibilidad local.
- `local`: Almacenamiento local y visibilidad local.



Tipos de variables

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para sistemas embebidos

Control de flujo

- `global`: Almacenamiento global y visibilidad global.
- `static`: Almacenamiento global y visibilidad local.
- `local`: Almacenamiento local y visibilidad local.



Tipos de variables

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para sistemas embebidos

Control de flujo

- `global`: Almacenamiento global y visibilidad global.
- `static`: Almacenamiento global y visibilidad local.
- `local`: Almacenamiento local y visibilidad local.



Modificadores de clase de almacenamiento

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para sistemas embebidos

Control de flujo

- **static**
- **volatile**
- **const**



Modificadores de clase de almacenamiento

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para sistemas embebidos

Control de flujo

- `static`
- `volatile`
- `const`



Modificadores de clase de almacenamiento

Clase 1

Introducción

Consideraciones
del Curso

Repaso de C

Motivación para
usar C

Consideraciones
de performance

C para sistemas embebidos

Control de flujo

- `static`
- `volatile`
- `const`

Programación de Microcontroladores

Clase 1

Carrera de Especialización en Sistemas Embebidos - FIUBA

Esp. Ing. Patricio Bos
pbos@fi.uba.ar

versión: 2016-08-20 rev 0.1