

Programa tentativo

Profs. Tomás S. Grigera y Jorge Runco

Unidad 1. Introducción a la programación: Octave/Matlab.

- 1.1. Algoritmos, programas y lenguajes. Datos e instrucciones. CPU, almacenamiento primario (memoria) y secundario (periféricos). Dispositivos de entrada y salida.
- 1.2. El intérprete Octave/Matlab. Expresiones aritméticas. Variables escalares, matriciales y alfanuméricas. Algunas funciones para manipulación de vectores y matrices. Polinomios. Definición de funciones mediante la construcción de función anónima. Gráficas simples de funciones.
- 1.3. Programas. Entrada y salida por consola. Estructuras de control: secuencia, decisión, iteración. Instrucciones `if`, `while`, `for`. Funciones. Entrada y salida a disco: `fopen`, `fprintf`, `fclose`. Gráficas de datos discretos. Ajuste de datos: `polyfit`.

Unidad 2. Representación de la información en sistemas digitales.

- 2.1. El bit. Almacenamiento de la información como colección de bits. Bytes, palabras y sus múltiplos.
- 2.2. Representaciones de números enteros. Bases para representación de números enteros positivos: números decimales, binarios, octales y hexadecimales. Enteros negativos: representación con signo y magnitud, inconvenientes. Representaciones en complemento a la base y complemento a la base reducida. Complemento a 2 y complemento a 1. Representación sesgada.
- 2.3. Representaciones de números reales. Representación de punto fijo. Representación de punto flotante: signo, exponente y mantisa. El standard IEEE 754. Aritmética en punto flotante, excepciones, `Inf` y `NaN`. Errores en la representación: error absoluto y relativo. Redondeo y truncamiento. Epsilon de la representación.
- 2.4. Representación de textos. Códigos de caracteres: BCD, ASCII, Unicode. Texto con formato: representación sobre texto ASCII o Unicode con “*markup*” (RTF, HTML, LaTeX) o binaria. Archivos de texto vs. binarios.
- 2.5. Representación de imágenes, sonido y video. Representación matemática de imágenes. Representación del color. Discretización: pixels y profundidad de color. Compresión con pérdida y sin pérdida. Formatos “*bitmap*”. Representación vectorial: SVG, Postscript. Representación matemática del sonido. Digitalización del sonido. Video.

Unidad 3. Elementos de arquitectura y organización de computadoras.

- 3.1. ¿Qué es una computadora? Computadoras analógicas y digitales. Arquitectura vs. organización.
- 3.2. Expresiones lógicas y álgebra de Boole. Circuitos digitales. Puertas lógicas: conjunto completo de puertas y programación de funciones arbitrarias. Circuitos integrados y PLAs. Circuitos combinatoriales: multiplexor, decodificador, sumador. La unidad lógica y aritmética (ULA). Relojes y circuitos secuenciales: latch y flip-flop. Memorias RAM estáticas y dinámicas. Contadores.
- 3.3. La unidad central de procesamiento (CPU) y la arquitectura de Von Neumann. Almacenamiento primario y secundario: memoria principal, periféricos, dispositivos de entrada, de salida y de entrada/salida. Buses: comunicación con memoria y periféricos. Organización de una CPU: ALU, registros, control y bus interno. El ciclo de ejecución y las interrupciones. Lenguaje de máquina: frontera software/hardware (Instruction Set Architecture). Microcódigo. Instrucciones típicas de un ISA.

- 3.4. El sistema operativo. Funciones del sistema operativo. Memoria virtual: páginas y fallas de página. Entrada/salida y control de periféricos (“*drivers*”). Procesos e hilos: contexto, alternancia de procesos, estados de un proceso. El cargador (“*bootstrap loader*”) y el BIOS.
- 3.5. Lenguaje ensamblador. La traducción o compilación. División de un programa ejecutable en módulos. Reubicación de módulos en memoria: el proceso de encadenado (“*linking*”) de módulos. Bibliotecas. Lenguajes de alto nivel.

Unidad 4. Programación en lenguaje C.

- 4.1. Introducción. Un primer programa. Proceso de compilación (programa fuente, programa objeto, bibliotecas, ejecutable). La función `main()` y entrada/salida elemental por consola. Argumentos de la línea de comando. Constantes numéricas, de caracteres y cadenas de caracteres. Expresiones: operadores aritméticos, lógicos, bit a bit, incremento y decremento. Asignación simple y asignación combinada con operación. Variables y tipos de datos. Definición de variables. Arreglos. Conversión entre tipos.
- 4.2. Funciones. La biblioteca `standard` y la biblioteca matemática. Estructuras de control: `if else`, `switch`, `while`, `do`, `for`. Interrupción de lazos y saltos: `continue`, `break`, `goto`.
- 4.3. Entrada/salida a archivos (`fopen`, `fclose`, `fprintf`, `fscanf`, `fgetc`, `fputc`, `fgets`). Entrada/salida sin formato (`fread` y `fwrite`). Tipo de datos `struct` (estructuras), `typedef` y `union`.
- 4.4. Declaración vs. definición de funciones: prototipos. Alcance o ámbito (scope) de variables. Variables globales. Módulos: funciones y variables externas. Símbolos públicos y privados: `static`. Variables automáticas y estáticas y segundo significado de `static`. Argumentos de funciones: noción de pasaje por valor y por referencia. Recursión.
- 4.5. Punteros. Punteros y arreglos. Cadenas de caracteres como arreglos de `char`. Algunas funciones para manipular cadenas. Asignación dinámica de memoria. Punteros a `struct`.

Unidad 5. Introducción al cálculo numérico.

- 5.1. Errores numéricos: errores de representación y de discretización. Propagación de errores.
- 5.2. Métodos básicos de aproximación de funciones. Polinomios. Evaluación numérica de polinomios. Interpolación. Polinomio interpolante de Lagrange. Polinomios osciladores. Interpolación con splines.
- 5.3. Ajuste de funciones. Ajuste por mínimos cuadrados como ajuste de máxima verosimilitud. Ajuste a una recta, a una función lineal en los parámetros y general
- 5.4. Derivación numérica: diferencias finitas, interpolación
- 5.5. Resolución de ecuaciones. Ecuaciones algebraicas de segundo y tercer grado. Raíces de polinomios. Raíces de funciones continuas: métodos de bisección, secante y *regula falsi*. Criterio de convergencia. Funciones derivables: método de Newton-Raphson. Puntos fijos: método de iteración, condiciones de validez. Sistemas de ecuaciones lineales.

Bibliografía

- Borrell i Nogueras, G. (2008), Introducción informal a Matlab y Octave, <http://webserver.dmt.upm.es/media/files/cursomo.pdf>.
- Eaton, J. W., Bateman, D. y Hauberg, S. (2008), *GNU Octave manual version 3*, Network Theory Ltd., 3 edition.
- González, R. C., Woods, R. E. y Eddins, S. L. (2009), *Digital image processing using Matlab*, Gatesmark publishing.
- Kernighan, B. W. y Ritchie, D. M. (1991), *El lenguaje de programación C*, Prentice-Hall Hispanoamericana, Mexico, 2 edition.
- Li, Z.-N. y Drew, M. S. (2004), *Fundamentals of multimedia*, Pearson Education International.
- Null, L. y Lobur, J. (2003), *Essentials of computer organization and architecture*, Jones and Bartlett, Sudbury, USA.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. y Flannery, B. P. (1992), *Numerical Recipes in C*, Cambridge University Press, 2 edition.

Sciutto, S. (2010), Apunte de cátedra, inédito.

Stallings, W. (2003), *Computer organization and architecture*, Prentice Hall, New Jersey.

Tanenbaum, A. S. (2006), *Structured computer organization*, Pearson Prentice Hall, 5 edition.

Web tutorial (2013), C programming web tutorial, http://www2.its.strath.ac.uk/courses/c/tableofcontents3_1.html.

Wikipedia (2013), Unicode, <http://en.wikipedia.org/wiki/Unicode>.

Sitio web

Sitio web de la asignatura (2016): <http://iflysib14.iflysib.unlp.edu.ar/tomas/docencia/computacion>