



UNIVERSIDAD BLAS PASCAL

Carrera: **INGENIERÍA EN INFORMÁTICA - 2003 - (Pres.)**

Asignatura: **PROGRAMACIÓN EFICIENTE**

Año: **2008**

PROPOSITO

Identificar las distintas problemáticas que afectan la performance de un software y adquirir los conocimientos necesarios para disminuir su impacto mediante técnicas de optimización de código y mediante el aprovechamiento del hardware disponible.

OBJETIVOS

- Encontrar las fuentes de ineficiencias de un programa.
- Ordenar las fuentes de ineficiencias según su impacto en la eficiencia global.
- Aplicar las facilidades del lenguaje de programación utilizado para mejorar la eficiencia del programa.
- Optimizar los bucles críticos reescribiéndolos para aprovechar al máximo las posibilidades del procesador y del sistema de memoria.
- Combinar varios algoritmos de manera balanceada, cuidando de que las optimizaciones locales no interfieran con la optimización global.
- Comprender las posibilidades avanzadas que ofrece el hardware para aumentar la eficiencia.
- Conocer herramientas para analizar la performance del código fuente.

CONTENIDOS

Unidad I: **ARQUITECTURA DE COMPUTADORAS MODERNAS**

Performance del software. Arquitecturas de microprocesadores. Tecnologías de memoria. Memorias cache. Precisión en números de punto flotante. Estándar IEEE 754.

Unidad II: **PROGRAMACIÓN Y PROFILING**

Rol del compilador. Optimización a cargo del compilador. Medición de tiempo de ejecución. Profiling mediante herramientas provistas en las distribuciones básicas. Profiling con herramientas más específicas. Detección de ineficiencias del uso de la memoria. Detección de no liberación de memoria.

Unidad III: **OPTIMIZACIÓN MANUAL DE CÓDIGO FUENTE**

Eliminación de acciones innecesarias. Function inlining. Optimización de ciclos de repetición. Patrones de acceso a memoria. Optimizaciones utilizando métodos matemáticos más eficientes. Uso eficiente de objetos.

Unidad IV: **SEGUIMIENTO DEL CÓDIGO**

Estudio de llamadas en un programa. Estudio de interacción entre objetos. Estudio de interacción entre hilos. Estudio de interacción de procesos en un programa paralelo.

BIBLIOGRAFIA

Básica:

- Kevin Dowd y Charles R. Severance. High Performance Computing. Segunda Edición. Ed.O'Reilly. (1998) - Isbn: 156592312X
- Kevin R. Wadleigh y Isom L. Crawford. Software Optimization for High-performance Computing. Ed.Prentice Hall. (2000) - Isbn: 0130170089
- Dov Bulka y David Mayhew. Efficient C++: Performance Programming Techniques. Ed.Addison-Wesley. (1999) - Isbn: 0-201-37950-3
- Jack Shirazi. Java Performance Tuning. Ed.O'Reilly. (2003) - Isbn: 0596003773
- David J. Lilja. Measuring Computer Performance: A Practitioner's Guide. Ed.Cambridge University Press. (2000) - Isbn: 0521641055

Complementaria:

- Scott Meyers. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. Tercera Edición. Ed.Addison Wesley. (2005) - Isbn: 0-201-92488-9
- Scott Meyers. More Effective C++: 35 New Ways to Improve Your Programs and Designs. Ed.Addison-Wesley. (1996) - Isbn: 0-201-63371-X

EVALUACIÓN

Regularización:

- La condición de alumno REGULAR se alcanzará aprobando 3 trabajos prácticos grupales consistentes en 1) el estudio de un software que se optimizará y corregirá en base a los temas abordados y 2) elaboración de un informe escrito.

Exámen final:

- Estudio de un caso, elaboración de un informe escrito y defensa oral.