

UNIVERSIDAD CARLOS III DE MADRID

Guía Presentación Bloque 5 (Temas 10 y 11)

Departamento de Ingeniería de Sistemas y
Automática

RAÚL PÉRULA MARTÍNEZ
LUIS ENRIQUE MORENO LORENTE
ALBERTO BRUNETE GONZALEZ
CESAR AUGUSTO ARISMENDI GUTIERREZ
DOMINGO MIGUEL GUINEA GARCIA ALEGRE
JOSÉ CARLOS CASTILLO MONTOYA



Universidad
Carlos III de Madrid



Esta obra se publica bajo una licencia CreativeCommons Reconocimiento-NoComercial-CompartidIgual 3.0
España.



Guíade presentación del bloque V (Tema 10. Introducción a la optimización de código; y tema 11. Riesgos de control y predicción de saltos)

Aunque las técnicas vistas en los bloques anteriores consiguen mejorar en gran medida las prestaciones de la CPU, existen casos en los que esto no es suficiente. Por este motivo, la optimización de código y la predicción de saltos juegan un papel crucial para conseguir optimizar al máximo la ejecución del código.

En esta línea, las técnicas de optimización de código se aplican después de la generación de código cumpliendo siempre la condición de que el código optimizado se debe comportar igual que el código original, excepto por ser más rápido o tener una longitud diferente. En el **tema 10** se describen tres técnicas clásicas de optimización: reordenación de código para eliminar detenciones en el cauce, desenrollamiento de bucles y software pipelining o segmentación software. Las técnicas basadas en reordenación de código intentan mejorar la localidad espacial de los programas para que el procesador esté ocupado la mayor parte del tiempo posible.

El desenrollamiento de bucles intenta aumentar el CPI de un programa al reducir (o eliminar) instrucciones de control del bucle y reduciendo la penalización por salto. Para esto, se duplica el cuerpo del bucle tantas veces como se quiera desenrollar. Esta técnica tiene especial efecto si se combina con la reordenación de código y el renombrado de registros.

La segmentación software es una técnicas utilizada para optimizar bucles que aprovecha las posibilidades de las arquitecturas que permiten ejecución fuera de orden. La principal característica de estas técnicas es el solapamiento entre iteraciones de un bucle siempre y cuando estas sean independientes. Esto maximiza el paralelismo además de reducir el tamaño del código respecto del resultado del desenrollado de bucles. Además, el cauce se llena y vacía solo una vez por bucle en vez de una vez por iteración desenrollada en caso de desenrollamiento.

Por otro lado, existe un tipo de riesgos que limitan en gran manera las prestaciones de un procesador. Esto son los llamados riesgos de control debidos a los saltos. El problema es conocer a qué dirección de memoria se debe saltar, y si el salto debe efectuarse o no. Para ello, en el **tema 11** se definen los tipos de saltos, atendiendo a las características anteriormente mencionadas. Aunque la solución más simple es detener el cauce hasta que se resuelvan las cuestiones anteriores, evidentemente no es eficiente. Por este motivo, se presenta una clasificación de las técnicas más conocidas para predicción de saltos,



Universidad Carlos III de Madrid
Departamento de Ingeniería de Sistemas y Automática

comenzando con las técnicas estáticas, las más simples, y finalizando con técnicas más avanzadas de predicción dinámica.