Instituto Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores
CE4301– Arquitectura de Computadores I
Johnny Zaet Agüero Sandí

Taller #3

1- Indique según la teoría ¿cómo se aplican las optimizaciones y para qué son? Ejecútelas y contraste la diferencia entre tres de ellas. Sugerencia: busque el término optimización gcc.

La optimización en compiladores, como GCC (GNU Compiler Collection), se refiere al proceso de mejorar el rendimiento y/o la eficiencia del código generado, sin cambiar su comportamiento externo. Las optimizaciones pueden mejorar la velocidad de ejecución, reducir el tamaño del código o ambos. GCC ofrece una amplia variedad de opciones de optimización que pueden ser utilizadas para adaptar el comportamiento del compilador a las necesidades específicas de una aplicación.

-O0 vs. -O2:

-O0: No realiza ninguna optimización. Es el nivel predeterminado y es útil durante la depuración, ya que no altera el flujo del código.

-O2: Realiza optimizaciones que no involucran un compromiso de espacio-tiempo. Puede mejorar significativamente el rendimiento sin aumentar mucho el tamaño del código.

-O2 vs. -O3:

-O2: Como se mencionó anteriormente, realiza optimizaciones sin comprometer el espacio.

-O3: Realiza todas las optimizaciones de -O2 y añade otras que pueden aumentar el tamaño del código. Es el nivel más agresivo de optimización.

-O2 vs. -Os:

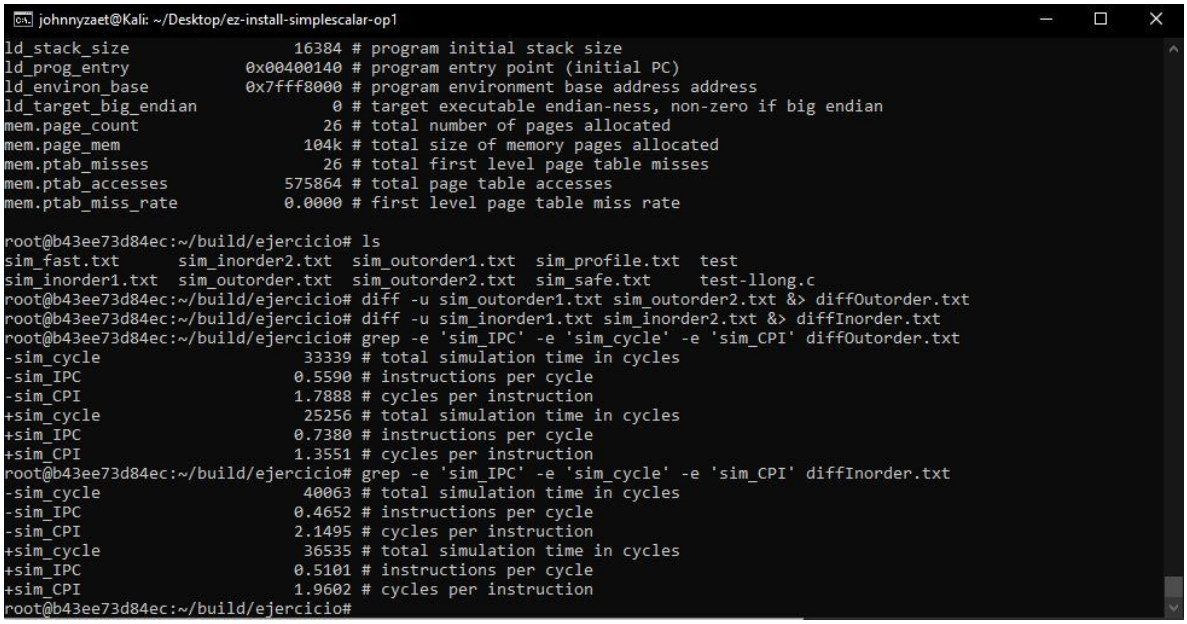
-O2: Optimiza el código para el rendimiento sin comprometer el espacio.

-Os: Optimiza el código para reducir su tamaño. Puede comprometer algunas optimizaciones de rendimiento para lograr un código más pequeño.

2- ¿Qué información le proporciona el simulador sim-fast sobre la aplicación?

La información proporcionada ofrece detalles sobre cómo se ejecutó una aplicación en un simulador, incluyendo estadísticas sobre la ejecución de instrucciones, la segmentación de memoria y las estadísticas relacionadas con el acceso y asignación de memoria.

3- Utilice el simulador sim-outorder por medio del uso de la opción -issue:inorder/ outoforder (defecto) , determine si la ejecución fuera de orden proporciona algún eneficio para la aplicación y explique según la teoría. Proporcione los resultados necesarios para sustentar su respuesta.



Como se puede observar en la imagen anterior, al utilizar el issue:inorder hace que el programa se ejecute más lento que fuera de orden.

4- El simulador sim-outorder le permite modificar el número de unidades funcionales, Explique sus resultados con base en el código fuente de la aplicación que se está analizando. Contraste el funcionamiento con respecto al archivo sin modificaciones.