

Compiladores

“¿Optimiza?”

Dragon Slayers

22 de noviembre de 2018

Interesa saber qué mejoras al código es capaz de efectuar su compilador preferido. Algunos tienen diversas opciones (o niveles) de optimización. Explore las relevantes. Al efecto, considere al menos las siguientes situaciones:

1. ¿Efectúa *constant folding*?
2. En GCC, es posible indicar que una función depende solo de sus argumentos, como por ejemplo el prototipo `__attribute__((pure)) int sqr(int);`. ¿Aprovecha realmente esto? O sea, si la función se llama varias veces con el mismo argumento, ¿la llama una sola vez?
3. ¿Expande funciones en línea según dirige por ejemplo el listado 1? ¿Lo hace automáticamente para funciones «simples» (como la anterior) si fue definida antes? ¿Si se define después?
4. ¿Se propagan valores de variables? O sea, por ejemplo si se asigna `a = 1;` y luego (después de varias otras líneas que no la afectan) se usa esta variable, ¿usa el valor? ¿Por ejemplo, elimina un `if (a != 1) { ... }`?
5. ¿Se recalculan (sub)expresiones que no varían? Considere por ejemplo el listado 2, la subexpresión `x * x` puede calcularse fuera del ciclo.
6. ¿Se calculan expresiones cuando se requieren solamente? Por ejemplo en el listado 3 el cálculo de `y` puede postergarse (e incluso evitarse según el valor de `flag`).
7. ¿Maneja *variables de inducción*, que tienen una relación lineal con el índice del ciclo? Por ejemplo, en el listado 4 podemos inicializar una variable temporal con `tmp = 15;` y en cada ciclo incrementar `tmp += 4;`.
8. En ciclos con cuerpo corto (como los ejemplos previos) el costo del ciclo puede ser una fracción relevante del costo total. La modificación de *loop unrolling* ejecuta varias iteraciones en cada ciclo.

Detalle el ambiente empleado (sistema operativo, arquitectura, compilador, opciones). Describa los experimentos que hace, incluyendo por cierto otros que agregue a las sugerencias, indicando cómo obtiene sus conclusiones. Dé un resumen general en un cuadro.

```
inline int sqr(int x)
{
    return x * x;
}
```

Listing 1: Función en línea

```
for(int y = 0; y < 10; y++)
    a[i] = x * x + y * y;
```

Listing 2: Subexpresión que no varía

```
y = ((3 * x + 20) * x - 135) * x + 42;
/* Rango de instrucciones que no usan y ni cambian x */
if(flag)
    printf("y = %lf", y);
/* Rango de instrucciones que no usan y */
```

Listing 3: Expresión que puede postergarse

```
x = 0;
for(i = 0; i < 200; i++)
    x += 4 * i + 15;
```

Listing 4: Variable de inducción

1. Condiciones de entrega

- La tarea se realizará *individualmente* (esto es grupos de una persona), sin excepciones.
- La entrega debe realizarse vía [Moodle](#) en un *tarball* en el área designada al efecto, bajo el formato `tarea-3-rol.tar.gz` (rol con dígito verificador y sin guión). Puede optar por otra compresión si Moodle la reconoce.

Dicho *tarball* debe contener las fuentes en LaTeX (al menos `tarea-3.tex`) de la parte escrita de su entrega, además de un archivo `tarea-3.pdf`, correspondiente a la compilación de esas fuentes.

- En caso de haber programas, su ejecutable *debe* llamarse `tarea-3`, de haber varias preguntas solicitando programas, estos deben llamarse `tarea-3-1`, `tarea-3-2`, etc. Si hay programas compilados, incluya una `Makefile` que efectúe las compilaciones correspondientes.

Los programas se evalúan según que tan claros (bien escritos) son, si se compilan y ejecutan sin errores o advertencias según corresponda. Parte del puntaje es por ejecución correcta con casos de prueba. Si el programa no se ciñe a los requerimientos de entrada y salida, la nota respectiva es cero.

- Además de esto, la parte escrita de la tarea debe en hojas de tamaño carta en Secretaría Docente de Informática (Piso 1, edificio F3).
- Tanto el *tarball* como la entrega física deben realizarse el día indicado en [Moodle](#). No entregar la parte escrita en papel o no entregar en formato electrónico tiene un descuento de 50 puntos.

Por cada día de atraso se descontarán 20 puntos. A partir del tercer día de atraso no se reciben más tareas, y la nota de la tarea es cero.

- Nos reservamos el derecho de llamar a interrogación sobre algunas de las tareas entregadas. En tal caso, la nota base (antes de descuentos por atraso y otros) es la de la interrogación. No presentarse a la interrogación sin justificación previa significa automáticamente nota cero.