

CC4302 Sistemas Operativos – Tarea 1 – Semestre Otoño 2024 – Profs.: Mateu, Torrealba, Arenas

Considere que Ud. viaja a Europa y puede llevar una maleta de hasta $maxW$ kilos. Dispone de un conjunto de n artículos $\{A_0, A_1, \dots, A_{n-1}\}$. El artículo A_i pesa $w[i]$ kilos y vale $v[i]$ euros. No puede llevar todos los artículos porque la suma de sus pesos excede $maxW$. Debe elegir qué artículos llevar maximizando la suma de sus valores. Este problema se conoce como Knapsack 0-1 y está en la categoría NP-difícil. El mejor algoritmo conocido que calcula la solución óptima toma demasiado tiempo: es $O(2^n)$. La función *llenarMaletaSec* de abajo es una heurística: entrega una buena solución para este problema y en un tiempo razonable, pero no es la solución óptima. Para lograrlo genera aleatoriamente k subconjuntos de artículos y elige el de mayor valor que no exceda $maxW$. Al retornar, **la solución del problema se entrega en el arreglo z** : el subconjunto con los artículos elegidos es $\{A_i \mid tq\ z[i]=1\}$. La función *random0or1* es dada y entrega aleatoriamente 0 o 1.

```
double llenarMaletaSec(double w[], double v[], int z[], int n,
                      double maxW, int k) {
    double best= -1;
    while (k-->0) {
        int x[n];
        double sumW= 0, sumV= 0;
        for (int i=0; i<n; i++) {
            x[i]= random0or1() && sumW+w[i]<=maxW ? 1 : 0;
            if (x[i]==1) {
                sumW += w[i];
                sumV += v[i];
            }
        }
        if (sumV>best) {
            best= sumV;
            for(int i=0; i<n; i++) {
                z[i]= x[i];
            }
        }
    }
    return best;
}
```

Programa la función *llenarMaletaPar* usando la misma heurística pero de modo que la elección se haga en paralelo para una máquina con 8 cores. Esta función recibe los mismos parámetros que *llenarMaletaSec*.

Metodología obligatoria: Lance 8 nuevos threads invocando 8 veces *pthread_create*. Cada thread evalúa $k/8$ subconjuntos aleatorios invocando *llenarMaletaSec(..., k/8)*, que está implementado en el archivo *test-maleta.c*. Cuidado: no puede compartir el arreglo z entre todos los threads. Debe crear un arreglo z independiente para cada thread. Use el thread principal solo para crear los threads y para elegir la mejor solución entre las mejores encontradas por los 8 threads. Si la

mejor solución fue por ejemplo la del thread 3, *llenarMaletaPar* debe retornar el valor *best* que calculó el thread 3 y copiar el arreglo z calculado por el thread 3 en el parámetro z de *llenarMaletaPar*. La forma de crear los threads es muy similar a la manera en que se crearon los threads para resolver en paralelo el problema de la búsqueda de un factor en la [clase auxiliar](#) del miércoles 12 de marzo.

Instrucciones

Baje *t1.zip* de U-cursos y descomprímalo. El directorio *T1* contiene los archivos *test-maleta.c*, *Makefile*, *maleta.h* (con los encabezados requeridos) y otros archivos. Ud. debe reprogramar en el archivo *maleta.c* la función *llenarMaletaPar*. Defina otras funciones si las necesita. **Se descontarán 5 décimas si su solución no usa la indentación de Kernighan.**

Pruebe su tarea bajo Debian 12. Ejecute el comando *make* sin parámetros. Le mostrará las opciones que tiene para compilar su tarea. Estos son los requerimientos para aprobar su tarea:

- *make run* debe felicitarlo por aprobar este modo de ejecución. El *speed up* reportado debe ser de al menos 1.5.
- *make run-g* debe felicitarlo.
- *make run-san* debe felicitarlo y no reportar ningún incidente en el manejo de memoria.

Cuando pruebe su tarea con *make run* en su notebook asegúrese de que posea al menos 2 cores, que esté configurado en modo alto rendimiento y que no estén corriendo otros procesos intensivos en uso de CPU al mismo tiempo. De otro modo podría no lograr el *speed up* solicitado.

Invoque el comando *make zip* para ejecutar todos los tests y generar un archivo *integral.zip* que contiene *integral.c*, con su solución, y *resultados.txt*, con la salida de *make run*, *make run-g* y *make run-san*.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *maleta.zip* generado por *make zip*. Recuerde descargar el archivo que subió, descargar nuevamente los archivos adjuntos y volver a probar la tarea tal cual como la subió a U-cursos. Solo así estará seguro de no haber entregado archivos incorrectos. No se consideran los días de receso, sábados, domingos o festivos.