

La función *integral* incluida en *test-integral.c* calcula secuencialmente la integral de una función que se recibe como parámetro. El encabezado de la función es:

```
typedef double (*Funcion)(void *ptr, double x);
double integral(Funcion f, void *ptr, double xi, double xf, int n);
```

*Integral* calcula numéricamente  $\int_{xi}^{xf} f(q, x) dx$  usando el método de los trapecios, en donde  $n$  es el número de trapecios usados para aproximar el área bajo la curva. Para ello se emplea la siguiente fórmula:

$$\int_{xi}^{xf} f(q, x) dx \approx h \cdot \left[ \frac{f(q, xi) + f(q, xf)}{2} + \sum_{k=1}^{n-1} f(q, xi + k \cdot h) \right]$$

con  $h = \frac{xf - xi}{n}$ . El puntero  $q$  se usa para pasar parámetros adicionales a la función en caso de necesidad. A modo de ejemplo suponga que Ud. dispone de la función  $g(x, y)$ . En el código de más abajo la función *integral\_g\_dx* usa *integral* para calcular numéricamente  $\int_{xi}^{xf} g(x, y) dx$ . Observe que como  $g$  no posee el tipo requerido por *integral*, se introduce *g\_aux* que sí posee el tipo requerido.

```
double g_aux(void *ptr, double x) {
    double y= *(double *)ptr;
    return g(x, y);
}

double integral_g_dx(double xi, double xf, double y, int n) {
    return integral(g_aux, &y, xi, xf, n);
}
```

Programa la función *integral\_par* con el siguiente encabezado:

```
double integral_par(Funcion f, void *ptr, double xi, double xf, int n, int p);
```

*Integral\_par* debe calcular la misma integral que la función *integral* de más arriba, pero en paralelo usando  $p$  cores. Para lograrlo use *fork* para crear  $p$  procesos pesados, descomponga el intervalo  $[xi, xf]$  en  $p$  subintervalos y llame en cada proceso pesado a *integral* (la versión secuencial) para calcular la integral de uno de los subintervalos, pero usando  $n/p$  trapecios. Se requiere que el incremento de velocidad (*speed up*) sea al menos un factor 1.7x. Cuando pruebe su tarea en su notebook asegúrese que posea al menos 2 cores y que esté configurado para el máximo desempeño. Si está configurado para ahorro de batería podría no lograr el *speed up* solicitado.

El programa de prueba incluido en *test-integral.c* calcula:  $\int_{yi}^{yf} \int_{xi}^{xf} \sin(x+2y) dx dy$  usando 10000 subintervalos para  $x$  y 20000 para  $y$ .

## Instrucciones

Descargue *t8.zip* de U-cursos y descomprímalo. Ejecute el comando *make* sin parámetros en el directorio *T8* para recibir instrucciones acerca del archivo en donde debe programar su solución (*integral.c*), cómo compilar y probar su solución y los requisitos que debe cumplir para aprobar la tarea.

## Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *integral.zip* generado por el comando *make zip*. Contiene los archivos *integral.c* y *resultados.txt*. A continuación es muy importante que descargue de U-cursos el mismo archivo que subió, luego descargue nuevamente los archivos adjuntos y vuelva a probar la tarea tal cual como la entregó. Esto es para evitar que Ud. reciba un 1.0 en su tarea porque entregó los archivos equivocados. Créame, sucede a menudo por ahorrarse esta verificación. Se descontará medio punto por día de atraso. No se consideran los días de receso, sábado, domingo o festivos.