

Se requiere calcular el producto de los enteros de un arreglo. El resultado puede ser muy grande y por lo tanto se usa el tipo *BigNum* que permite representar enteros positivos de tamaño arbitrario. Ya existe la función *seqArrayProd*, programada en *test-prod.c*, que calcula secuencialmente el producto usando divide y conquista. Su código es similar a este:

```
BigNum *seqArrayProd(int a[], int i, int j) {
    if (i==j) {
        return smallNum(a[i]); // Convierte un entero de C a BigNum
    }
    else {
        int h= (i+j)/2;
        BigNum *left= seqArrayProd(a, i, h);
        BigNum *right= seqArrayProd(a, h+1, j);
        // Multiplicacion de BigNum's
        BigNum *prod= bigMul(left, right);
        freeBigNum(left); // Hay que liberar la memoria
        freeBigNum(right); // ocupada por los BigNum's
        return prod;
    }
}
```

Ud. debe programar en el archivo *prod.c* la función:

*BigNum *parArrayProd(int a[], int i, int j, int p);*

Esta función debe calcular el producto de los enteros $a[i]$, $a[i+1]$, ..., $a[j]$ usando divide y conquista en paralelo con p threads. Deberá crear los threads con la función *pthread_create*. La solución es similar a la de quicksort paralelo. Cuando p sea 1, Ud. debe invocar la función *seqArrayProd* que hace el cálculo en un solo thread, secuencialmente. Lea las observaciones en la plantilla *prod.c.plantilla*, le ayudará a resolver este problema.

Instrucciones

Baje *t1.zip* de U-cursos y descomprímalo. El directorio *T1* contiene los archivos *test-prod.c*, *Makefile*, *prod.h* (con los encabezados requeridos) y otros archivos. Ud. debe programar en el archivo *prod.c* la función *parArrayProd*.

Pruebe su tarea bajo Debian 11 de 64 bits. 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 1.2.

- *make run-g* debe felicitarlo.
- *make run-san* debe felicitarlo.

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 *prod.zip* que contiene *prod.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 *prod.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, sábado, domingo o festivos.