

CC3301 Programación de Software de Sistemas – Semestre Primavera 2022 – Tarea 1 – Prof.: Luis Mateu

Programa eficientemente la siguiente función:

```
typedef unsigned int uint; // enteros sin signo
uint concat_bits(uint x, int n, uint y, int m);
```

Si los bits de x son $x_{31} x_{30} \dots x_1 x_0$ en donde x_0 es el bit menos significativo y los de y son $y_{31} y_{30} \dots y_1 y_0$, la función `concat_bits` debe retornar $0 \dots 0 x_{n-1} \dots x_0 y_{m-1} \dots y_0$. Considere que $0 < n, m < 32$ y que $n+m \leq 32$. Ejemplos de uso:

```
uint z1= concat_bits(0b101, 3, 0b11001, 5);
// z1=0b10111001
uint z2= concat_bits(0b11011, 3, 0x10101, 2);
// z2=0x01101
```

Restricciones:

- Ud. no puede usar los operadores de multiplicación, división o módulo ($*$ / $\%$). Use los operadores de bits eficientemente.
- Si necesita calcular el número de bits en variables de tipo `uint`, calcule `sizeof(uint)<<3`. La cantidad de bits en un byte es siempre 8.
- Se descontará medio punto por no usar el estilo de indentación de Kernighan como se explica en [esta sección](#) de los apuntes.
- El estándar de C no especifica el resultado para desplazamientos mayores o iguales al tamaño del operando. Si desplaza un entero x de 32 bits en 32 o más bits, no pasará la prueba con `sanitize` (`make run-san`).

Instrucciones

Baje `t1.zip` de U-cursos y descomprímalo. El directorio `T1` contiene los archivos (a) `test-concat.c` que prueba si su tarea funciona y compara su eficiencia con la solución del profesor, (b) `prof.ref` con el binario ejecutable de la solución del profesor, (c) `concat.h` que incluye el encabezado de la función pedida, y (d) `Makefile` que le servirá para compilar y ejecutar su tarea. Ud. debe programar la función `concat_bits` en el archivo `concat.c`. Se incluye una plantilla en `concat.c.plantilla`.

Pruebe su tarea bajo Debian 11 de 64 bits nativo o virtualizado con VirtualBox. Los usuarios de Mac OS X pueden virtualizar con VMware o UTM/Qemu. **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. Su solución no debe ser 80% más lenta que la solución del profesor.
- `make run-g` debe felicitarlo.
- `make run-san` debe felicitarlo y no reportar ningún problema como por ejemplo desplazamientos indefinidos.

Cuando pruebe su tarea con `make run` asegúrese que su computador 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 la eficiencia solicitada.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo `concat.zip` generado por el comando `make zip`. **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.