

CC3301 Programación de software de sistemas – Tarea 5 – Primavera 2024 – Profesores Mateu/Ibarra/Urrea

La función *sort* está programada en C y en assembler Risc-V en los archivos *sort-c.c* y *sort-rv.s*. Esta función ordena ascendentemente un arreglo *nums* de *n* enteros sin signo usando un algoritmo ridículamente ineficiente, pero muy simple. El código equivalente en C está comentado, mostrando la ubicación de las variables en los registros.

El encabezado de la función es: *void sort(unsigned int nums[], int n);*

Los archivos *sort-c-hexamax.c* y *sort-rv-hexamax.s* son copias de *sort-c.c* y *sort-rv.s* respectivamente. Modifique las funciones *sort* en *sort-c-hexamax.c* y *sort-rv-hexamax.s* de modo que se ordene el arreglo descendientemente según la máxima cifra en hexadecimal. El siguiente ejemplo muestra el ordenamiento ascendente versus el ordenamiento solicitado.

Orden ascendente normal	Orden descendente solicitado
0x0	0x123f
0x1a	0x1a
0x45	0x45
0x123f	0x100000
0x100000	0x0

Los números están en hexadecimal para que sea más fácil determinar la cifra hexadecimal de mayor valor. Por ejemplo la cifra máxima de 0x1a es a (10) y la de 0x45 es 5. En la columna derecha, la máxima cifra hexadecimal es f (15), a (10), 5, 1 y 0.

Instrucciones

Baje *t5.zip* de U-cursos y descomprímalo. Contiene el *Makefile* y los archivos que necesita para hacer esta tarea. Ejecute el comando *make* sin parámetros para recibir instrucciones sobre la ubicación de los archivos que debe modificar y cómo compilar, ejecutar y depurar. En particular lea los tips para la depuración y la solución de problemas.

Restricciones

No hay restricciones para la programación de *sort-c-hexamax.c*. Para la programación de *sort-rv-hexamax.s* Ud. **solo puede modificar el código que compara los elementos consecutivos**. Está claramente delimitado en el archivo original. No modifique nada más. Sin esta restricción la tarea sería trivial. **Además para hacer la comparación no puede invocar otras funciones**. Una vez hecha la comparación, la ejecución debe continuar en la etiqueta *.decision* y el resultado de la comparación debe quedar en el registro *t1*. Si $t1 > 0$, los números en *p[0]* y *p[1]* están desordenados y por lo tanto se intercambiarán. Si

$t1 \leq 0$ no se intercambiarán.

Ayuda

- Modifique en *sort-c-hexamax.c* la función *sort* de manera que se comparen los números por la cifra hexadecimal mayor. Haga un esfuerzo en llegar a la función más pequeña, porque así será menor la cantidad de líneas en assembler que deberá programar y depurar. Pruebe *sort-c-hexamax.c* ejecutando el comando *make sort-c-hexamax.run* (*make sort-c-hexamax.ddd* para depurar)
- Una vez que pase exitosamente la prueba de *sort-c-hexamax.c*, ejecute el comando *make sort-c-hexamax.s* para compilar a assembler. Estudie en el archivo *sort-c-hexamax.s* la traducción a assembler Risc-V de la función que compara los números. Use un código similar para comparar los números en la zona delimitada en *sort-rv-hexamax.s* y así completar su tarea. Recuerde: sin invocar otras funciones. Pruebe su tarea con: *make sort-rv-hexamax.run* (*make sort-rv-hexamax.ddd* para depurar)
- Revise si el toolchain para compilar y ejecutar programas para Risc-V está instalado en */opt/riscv*. Si no encuentra ese directorio, descargue el archivo *riscv.tgz* (*riscv-arm.tgz* para Arm) de [esta carpeta de google drive](#) e instale el *toolchain* para RiscV con estos comandos:

```
cd ... directorio en donde descargó riscv.tgz ...
sudo bash
cat riscv.tgz | ( cd / ; tar zxvf - )
```
- En la clase auxiliar del viernes 4 de octubre se estudió la solución de una tarea similar de un semestre pasado.

Entrega

Entregue por medio de U-cursos el archivo ***hexamax.zip*** generado con el comando *make zip*. Este comando verifica que su tarea funcione correctamente y luego genera *hexamax.zip* con los archivos *sort-c-hexamax.c*, *sort-rv-hexamax.s* y *resultados.txt* con la ejecución de la tarea. **Recuerde descargar de u-cursos lo que entregó, descargar 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. Su tarea debe ordenar correctamente, si no será rechazada. Se descontará medio punto por día de atraso (excluyendo sábados, domingos, festivos o vacaciones).