

CC3301 Programación de software de sistemas – Tarea 5 – Primavera 2023 – Profesores Mateu/Ibarra

Una palabra es una secuencia contigua de caracteres que no contiene ningún espacio en blanco. Por ejemplo el string "estaesunasolapalabra," contiene una palabra y el string " estas son 4 palabras. " contiene 4 palabras.

La función *sort* está programada en C en el archivo *sort-c.c* y en assembler Risc-V en *sort-rv.s*. Esta función ordena lexicográficamente un arreglo de strings usando un algoritmo ridículamente ineficiente, porque toma tiempo $O(n^3)$. En *sort-rv.s*, el código equivalente en C está comentado, mostrando la ubicación de las variables en los registros.

El encabezado de la función es: `void sort(char *a[], int n);`

En el arreglo de strings *a* vienen *n* frases como "quien quiere queque" o "antes yo no era asi y ahora sigo igual".

El archivo *sort-rv-wc.s* es una copia de *sort-rv.s* y *sort-c-wc.c* es una copia de *sort-c.c*. Modifique las funciones *sort* en *sort-c-wc.c* y en *sort-rv-wc.s* de modo que se ordene el arreglo de strings ascendentemente por el número de palabras en el string. El siguiente ejemplo muestra a la izquierda un arreglo ordenado lexicográficamente y a la derecha ordenado por número de palabras.

ordenado lexicográficamente	ordenado por número de palabras
Defina universo y de 2 ejemplos El futuro ya no es lo que solía ser Soy inseguro, ¿o no?	Soy inseguro, ¿o no? Defina universo y de 2 ejemplos El futuro ya no es lo que solía ser

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-wc.c*. Para la programación de *sort-rv-wc.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

- Programe en *sort-c-wc.c* una función auxiliar que entregue el número de palabras en un string. 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-wc.c* ejecutando el comando: *make sort-c-wc.run* (*make sort-c-wc.ddd* para depurar)
- Una vez que pase exitosamente la prueba de *sort-c-wc.c*, ejecute el comando *make sort-c-wc.s* y luego estudie en el archivo *sort-c-wc.s* la traducción de la función que calcula el número de palabras a assembler Risc-V. Use un código similar para calcular el número de palabras en la zona delimitada en *sort-rv-wc.s* y así completar su tarea. Recuerde: sin invocar otras funciones. Pruebe su tarea con: *make sort-rv-wc.run* (*make sort-rv-wc.ddd* para depurar)
- Revise si el toolchain para compilar y ejecutar programas para RiscV 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 6 de octubre se estudió la solución de una tarea similar de un semestre pasado.

Entrega

Entregue por medio de U-cursos el archivo *wc.zip* generado con el comando *make zip*. Este comando verifica que su tarea funcione correctamente y luego genera *wc.zip* con los archivos *sort-c-wc.c*, *sort-rv-wc.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).