

Universidad de Buenos Aires, F.I.U.B.A.
66.20 Organización de Computadoras
Trabajo práctico 1: conjunto de instrucciones MIPS
2^{do} cuatrimestre de 2013

\$Date: 2013/11/07 02:34:04 \$

1. Objetivos

Familiarizarse con el conjunto de instrucciones MIPS y el concepto de ABI, extendiendo un programa que resuelva el problema descrito en la sección 4.

2. Alcance

Este trabajo práctico es de elaboración grupal, evaluación individual, y de carácter obligatorio para todos alumnos del curso.

3. Requisitos

El informe deberá ser entregado personalmente, por escrito, en la fecha estipulada, con una carátula que contenga los datos completos de todos los integrantes.

Además, es necesario que el trabajo práctico incluya (entre otras cosas, ver sección 6), la presentación de los resultados obtenidos, explicando, cuando corresponda, con fundamentos reales, las causas o razones de cada caso.

4. Descripción

En este trabajo, se reimplementará parcialmente en assembly MIPS el programa desarrollado en el trabajo práctico anterior [1].

Para esto, se requiere reescribir el programa, de forma tal que quede organizado de la siguiente forma:

- **main.c**: contendrá todo el código necesario para el procesamiento de las opciones de línea de comandos, apertura y cierre de archivos (de ser necesario), y reporte de errores (**stderr**). Desde aquí se llama a la función que invierte líneas siguiente.
- **reverse.S**: contendrá el código MIPS32 assembly con la función **reverse()**, y las funciones y estructuras de datos auxiliares que los alumnos crean convenientes (ej: para reserva de memoria). También contendrá la definición en assembly de un vector equivalente al siguiente vector C: **const char* reverse_errmsg[]**. Dicho vector contendrá los mensajes de error que las funciones antes mencionadas puedan generar, y cuyo índice es el código de error devuelto por las mismas.
- Los header files pertinentes (al menos, **reverse.h**, con el prototipo de **reverse()**, a incluir en **main.c**) y la declaración del vector **extern const char* reverse_errmsg[]**¹.

A su vez, el prototipo C de la función MIPS32 **reverse()** es el siguiente:

- **int reverse(int infd, int outfd)**

La función recibe por **infd** y **outfd** los file descriptors correspondientes a los archivos de entrada y salida pre-abiertos por **main.c**.

Ante un error, ambas funciones volverán con un código de error numérico (índice del vector de mensajes de error de **reverse.h**), o cero en caso de realizar el procesamiento de forma exitosa.

5. Implementación

El programa a implementar deberá satisfacer algunos requerimientos mínimos, que detallamos a continuación:

5.1. ABI

Será necesario que el código presentado utilice la ABI explicada en clase ([2] y [3]).

5.2. Syscalls

Es importante aclarar que desde el código assembly no podrán llamarse funciones que no fueran escritas originalmente en assembly por los alumnos (o las provistas por la cátedra). Por lo contrario, desde el código C sí podrá (y deberá) invocarse código assembly.

¹no confundir con la definición, que deberá implementarse en assembly dentro de **reverse.S**

Por ende, y atendiendo a lo planteado en la sección 4, los alumnos deberán invocar algunos de los system calls disponibles en NetBSD (en particular, `SYS_read` y `SYS_write`).

5.3. Casos de prueba

Es necesario que la implementación propuesta pase todos los casos incluidos tanto en el enunciado del trabajo anterior [1] como en el conjunto de pruebas suministrado en el informe del trabajo, los cuales deberán estar debidamente documentados y justificados.

5.4. Documentación

El informe deberá incluir una descripción detallada de las técnicas y procesos de desarrollo y debugging empleados, ya que forman parte de los objetivos principales del trabajo.

6. Informe

El informe deberá incluir:

- Este enunciado;
- Documentación relevante al diseño, desarrollo y debugging del programa;
- Las corridas de prueba, (sección 5.3) con los comentarios pertinentes;
- El código fuente completo, en dos formatos: digitalizado² e impreso en papel.

7. Fechas

La fecha de la primera oportunidad de entrega, es el martes 19/11. La fecha de vencimiento (y fin del curso) es el martes 3/12.

Referencias

- [1] Enunciado del primer trabajo práctico (TP0), segundo cuatrimestre de 2013 (<http://groups.yahoo.com/groups/orga-comp/files/TPs/>).
- [2] System V application binary interface, MIPS RISC processor supplement (third edition). Santa Cruz Operations, Inc.

²No usar diskettes: son propensos a fallar, y no todas las máquinas que vamos a usar en la corrección tienen lectora. En todo caso, consultá con tu ayudante.

- [3] MIPS ABI: Function Calling Convention, Organización de computadoras - 66.20 (archivo "func_call_conv.pdf", <http://groups.yahoo.com/groups/orga-comp/Material/>).