

Trabajo Practico 1: Conjunto de instrucciones MIPS

Augusto Arturi (#97498)
turitoh@gmail.com

Matias Rozanec (#97404)
rozanecm@gmail.com

Agustin Miguel Payaslian (#96.885)
payas17@hotmail.com

Grupo Nro. # - 2do. Cuatrimestre de 2017

66.20 Organización de Computadoras
Facultad de Ingeniería, Universidad de Buenos Aires

7/09/2017

Resumen

El trabajo consiste en programar la Criba de Eratostenes, con el objetivo de familiarizarse con las herramientas que utilizaremos a lo largo del curso.

1. Introducción

Aquí se comenta en forma escueta como está constituido el presente informe, donde básicamente se encuentran dos secciones principales: Desarrollo y Conclusiones.

En Desarrollo se encuentran breves comentarios sobre la implementación del algoritmo como también, las corridas de prueba del programa. En la sección conclusiones se discuten los resultados obtenidos.

2. Desarrollo

2.1. Implementación

2.2. Corridas de prueba

3. Conclusiones

Como hemos visto en los ejercicios anteriores, con los problemas de seguridad que presentan los programas compilados en C, se puede modificar el valor de una variable con el hecho de ingresar un string. Esto no es un problema menor ya que así como modificamos el valor de una variable, también se puede llamar a una función dentro del programa o hasta se pueden ingresar instrucciones. Esto puede llegar a generar inconvenientes a la hora de tener un programa con ingreso de usuario y contraseña o cualquier otro en el que se necesite el ingreso de datos, ya que se puede escribir código malicioso por la entrada manipulando el programa de la forma deseada.

Una posible solución a este problema es chequear la cantidad de caracteres que se están ingresando y cortar la entrada una vez alcanzado el tamaño del buffer.

A. Apendice A: Código fuente

B. Apendice B: Enunciado

66:20 Organización de Computadoras

Trabajo práctico 0: Infraestructura básica

1. Objetivos

Familiarizarse con las herramientas de software que usaremos en los siguientes trabajos, implementando un programa (y su correspondiente documentación) que resuelva el problema piloto que presentaremos más abajo.

2. Alcance

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

3. Requisitos

El trabajo deberá ser entregado personalmente, 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 7), la presentación de los resultados obtenidos, explicando, cuando corresponda, con fundamentos reales, las causas o razones de cada resultado obtenido.

El informe deberá respetar el modelo de referencia que se encuentra en el grupo, y se valorarán aquellos escritos usando la herramienta \TeX / \LaTeX .

4. Recursos

Usaremos el programa GXemul [1] para simular el entorno de desarrollo que utilizaremos en este y otros trabajos prácticos, una máquina MIPS corriendo una versión relativamente reciente del sistema operativo NetBSD [2].

5. Programa

Se trata de una versión en lenguaje C de la Criba de Eratóstenes [3]. El programa recibirá por `stdin` un número natural N , y dará por `stdout` (o

escribirá en un archivo) una lista de todos los números primos menores que N . De haber errores, los mensajes de error deberán salir exclusivamente por `stderr`.

5.1. Comportamiento deseado

Primero, usamos la opción `-h` para ver el mensaje de ayuda:

```
$ erat -h
Usage:
  erat -h
  erat -V
  erat [options] N
Options:
  -h, --help      Prints usage information.
  -V, --version   Prints version information.
  -o, --output    Path to output file.
Examples:
  erat -o - 10
```

Ahora usaremos el programa para obtener los números primos menores que 10. Usamos “-” como argumento de `-o` para indicarle al programa que imprima el tablero por `stdout`:

```
$ (echo 10) | erat
2
3
5
7
```

El programa deberá retornar un error si su argumento está fuera del rango $[2, \text{MAXINT}]$.

6. Implementación

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

El propósito de `erat()` es simple: eliminar, de un arreglo de números naturales consecutivos que comienza por 2, todos los números que no sean primos.

```
void erat(unsigned int *p, int n);
```

El programa deberá procesar los argumentos de entrada, crear el arreglo de números naturales consecutivos, de los cuales `erat()` debe poner en 0 los que no sean primos, y escribir en `stdout` o un archivo aquellos que hayan resultado ser primos.

6.1. Algoritmo

El algoritmo a implementar es la Criba de Eratóstenes [3], explicado en clase.

7. Informe

El informe deberá incluir:

- Este enunciado;
- Documentación relevante al diseño e implementación del programa;
- Corridas de prueba para $N = -5, 1, 10, 50$ y 100 , con los comentarios pertinentes;
- El código fuente completo, en dos formatos: digitalizado e impreso en papel.

8. Fecha de entrega

La última fecha de entrega y presentación es el jueves 7 de Septiembre de 2017.

Referencias

- [1] GXemul, <http://gavare.se/gxemul/>.
- [2] The NetBSD project, <http://www.netbsd.org/>.
- [3] Criba de Eratóstenes, http://http://es.wikipedia.org/wiki/Criba_de_Erat%C3%B3stenes.