66.20 Organización de Computadoras
Trabajo Práctico 0:
Infraestructura básica

Burdet Rodrigo, *Padrón Nro. 93440*
rodrigoburdet@gmail.com

Romani Nazareno, *Padrón Nro. 83991*
nazareno.romani@gmail.com

Martinez Gaston Alberto, *Padrón Nro. 91383*
gaston.martinez.90@gmail.com

1er. Cuatrimestre de 2014
66.20 Organización de Computadoras
Facultad de Ingeniería, Universidad de Buenos Aires

30/03/2014

1. Objetivos

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

2. Resumen

En el presente trabajo, se implementó un algoritmo que resuelve la transformación de un conjunto arbitrario de bytes en un conjunto formado por caracteres ASCII y viceversa. Los distintos valores a codificar/decodificar, son obtenidos a través de los parámetros definidos en el enunciado. El programa fue compilado tanto en la máquina host (sistema operativo linux), como en una máquina corriendo el sistema operativo NetBSD.

3. Desarrollo

3.1. Paso 1: Configuración de Entorno de Desarrollo

El primer paso fue configurar el entorno de desarrollo, de acuerdo a la guía facilitada por la cátedra.
Trabajamos con la distribución de linux basada en Debian y con el GxEmul proporcionado por la cátedra, el cual tiene ya configurado NetBSD.

3.2. Paso 2: Implementación del programa

El programa debe ejecutarse por línea de comando y la salida del mismo dependerá del valor de los argumentos con los que se lo haya invocado.

3.2.1. Ingreso de parámetros

El formato para invocar al programa es el siguiente:

./tp0 [OPTIONS]

Los parámetros válidos que puede recibir el programa son los siguientes:

-e , –**encode** (Encodes to Base64).
-d , –**decode** (Decodes from Base64).
-i , –**input file** (Reads from file or stdin).
-o , –**output file** (Writes to file or stdout).
-v , –**version** (Show version string).
-h , –**help** (Print this message and quit).

En caso de recibir un parámetro diferente a alguno de los listados anteriormente, la salida del programa será la misma que la de la opción -h, para despejar posibles dudas de uso. VER VER

3.2.2. Interpretación de parámetros

Para parsear los parámetros se usaron las funciones definidas en arg_parse.h. Se puede conocer más en detalle el funcionamiento de las mismas, a través de la documentación incluida en dicho archivo. Estas funciones permiten recoger los parámetros de entrada del programa y ejecutar la funcionalidad correspondiente. Estas son compatibles con NetBSD.

4. Código para compilar el programa con gcc

Para poder compilar el proyecto, se debe abrir una terminal linux dentro del directorio donde se encuentra el código fuente escrito en C, y utilizar el siguiente comando:

```
gcc -o tp0 tp0.c
```

Esto generara un archivo ejecutable, llamado tp0, con el que luego realizaremos las pruebas.
Para generar el cógido en MIPS32, generado por el compilador, vamos a usar el siguiente comando:

gcc -march=mips32 -S -c tp0.c -o tp0.S

5. Corridas de prueba y Mediciones

En las figuras que siguen a continuación se muestran los comandos utilizados para ejecutar el programa y se puede apreciar en los gráficos los resultados de las diferentes pruebas que realizamos.

Figura 1: Corrida 1

Figura 2: Corrida 2

Figura 3: Corrida 3

6. Conclusiones

Como se enuncia en el objetivo de este trabajo práctico, aprendimos a instalar y manejar el GxEmul, a realizar transferencias de archivos en linux, así como también compilar y ejecutar programas en el NetBSD. Por otro lado, aprendimos a manejar y escribir informes en LaTeX. De este modo, estamos preparados para que en los próximos trabajos prácticos, nos aboquemos directamente al desarrollo de los mismos.