

Proyecto 4: Compactador de Archivos

Jueves 24 de Octubre

I. DESCRIPCIÓN GENERAL

Este programa compacta todos los archivos que se le indiquen, usando el código Huffman generado por el Proyecto previo. Al final se despliegan algunas estadísticas básicas. Es muy importante el manejo de argumentos de la línea de comando para este proyecto. Toda la programación debe hacerse en Lenguaje C sobre Linux.

II. PRELIMINARES

El proyecto previo generó un archivo `.h` con el código Huffman binario correspondiente a cada uno de los 256 caracteres de ASCII extendido. El programa desarrollado en este proyecto incluirá dicha tabla (*i.e.*, `#include`) al ser compilado. Nótese que se compila una sola vez y que de ese punto en adelante se usa repetidamente el programa compilado. Si el `.h` no fuera conveniente, pueden modificar el proyecto previo según necesiten.

III. ARGUMENTOS

Su programa (identificado como `<prog>`) se invocará de esta manera:

```
<prog> <file-list>
```

Se procesarán todos los archivos mencionados en `<file-list>` y cada uno será compactado.

Ejemplos:

- `<prog> W B C D`
- `<prog> *.c`
- `<prog> *.c *.tex`
- `<prog> *`
- `<prog> W.c`

IV. SALIDA

Por cada archivo a ser compactado se creará (o se reutilizará) un archivo con el mismo nombre que el archivo original al que se le añade la extensión `".huff"`. El archivo original no se pierde.

Dicho archivo tendrá un *header* de 3 bytes que indicará la cantidad de caracteres ASCII que el archivo original tenía. Esta información se graba en binario. Esto limita el tamaño de los archivos que se pueden compactar a 2^{24} bytes. Inmediatamente aparecen todos los **bits** que corresponden a cada byte del archivo original de acuerdo al código Huffman que

estemos usando. Obviamente, la frontera entre un símbolo y el siguiente puede ocurrir dentro de un byte. Se deben hacer manipulaciones de bits para obtener el resultado correcto.

Después de procesar todos los archivos de la lista, su programa despliega en consola una tabla que muestre para cada archivo su tamaño original, su nuevo tamaño y el porcentaje de compactación logrado. Si algún archivo crece se debe indicar esta situación. Si la lista incluye más de un archivo, se debe dar al final un gran total de compactación: bytes de entrada, bytes de salida y porcentaje de compactación¹.

V. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto "no revisable" y recibe un 0 de calificación inmediata:

- La colaboración entre grupos se considera fraude académico.
- Todo el código debe estar escrito en C (no C++).
- La única interacción válida con el programa es por medio de argumentos de línea de comando.
- El proyecto debe compilar y ejecutar en Linux. Todo debe estar **integrado**, explicaciones del tipo "*todo está bien pero no pudimos pegarlo*"² provocan la cancelación automática de la revisión.
- La presentación debe ser de mucha calidad.
- No debe dar "Segmentation Fault" bajo ninguna circunstancia.
- Hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.

VI. FECHA DE ENTREGA

Revisiones a las 11:30 am el **Jueves 24 de Octubre** en la oficina del profesor. Mande además un `.tgz` con todo lo necesario (fuentes, `makefile`, `readme`, etc.) a `torresrojas.cursos.05@gmail.com`. Ponga como subject: "[AA] - Proyecto 4 - Fulano - Mengano - Sutano", donde Fulano, Mengano y Sutano son los miembros del grupo.

¹Incluyan los 3 bytes del header en todos sus cálculos

²esto incluye los supuestos casos cuando alguien del grupo de trabajo no hizo su parte – el profesor no está interesado en sus problemas de organización.