

LZ4

3 de mayo de 2021

[LZ4](#) es un algoritmo de compresión sin pérdidas que usa técnicas de diccionario de la familia del LZ77 propuesto por Yann Collet en 2011.

Está implementado en el núcleo de Linux y es usado por algunos *filesystem* para compresión *on-the-fly*. En la página citada anteriormente hay un apartado dedicado a mostrar algunos de sus usos.

La práctica consistirá en implementar en PHYTON 3 una aplicación¹ que sea capaz leer/escribir bloques de datos compatibles con la especificación:

[LZ4 Block Format Description](#)

Requisitos

La aplicación ha de ser capaz de:

1. leer un fichero de cualquier tipo² y crear otro fichero conforme con [LZ4 Block Format](#),
2. leer un fichero conforme con [LZ4 Block Format](#) y recuperar el fichero original.

En todos los casos, el tiempo T de ejecución para un fichero ha de ser inferior³ a los 300s y la ratio de compresión $R = \frac{\text{Tamaño fichero original}}{\text{Tamaño fichero comprimido}} \geq 1,1$.⁴

¹Por supuesto, no se pueden usar *packages/libraries* que ya implementen el formato.

²El tamaño máximo será de 4 Mbytes.

³Este límite es para evitar que se alargue en demasía la ejecución en busca de una mayor ratio de compresión.

⁴Este límite es para evitar que en aras de una mayor rapidez la ratio de compresión pueda llegar a ser menor que 1.

Evaluación

Se permitirá la entrega de una versión para cada una de las categorías:

- a) optimizada para la ratio de compresión R ,
- b) optimizada para el tiempo de ejecución T y $R \geq 1,1$
- c) optimizada para la transmisión (Tiempo de compresión, transmisión y descompresión).

La nota de la práctica estará compuesta por:

- 62,5 % si funciona correctamente.
- 2,5 % por cada categoría si la versión entregada es mejor que la implementación de referencia del profesor.
- Por cada categoría se establecerá un ranking que determinará qué parte del 10 % de la nota correspondiente a cada categoría se adjudicará.

Entrega

Tres ficheros⁵:

- `lz_R.py` optimizado para la ratio de compresión R ,
- `lz_T.py` optimizado para el tiempo de ejecución T y $R \geq 1,1$,
- `lz.py` optimizado para la transmisión (Tiempo de compresión, transmisión y descompresión).

La sintaxis de ejecución ha de ser compatible con:

- > `python lz_T.py -c nombre_fichero`
siendo `nombre_fichero` un fichero cualquiera; el resultado será un fichero conforme con [LZ4 Block Format](#) con nombre `nombre_fichero.lz4`.
- > `python lz_T.py -d nombre_fichero.lz4`
siendo `nombre_fichero.lz4` un fichero conforme con [LZ4 Block Format](#); el resultado será un fichero que coincida con el original al que se le ha aplicado el [LZ4 Block Format](#) y con nombre `nombre_fichero`.

⁵No se admitirán ficheros con otros nombres.

Análogamente para el resto de las versiones.

La fecha límite de entrega serán las **23:59 del martes 1 de junio de 2021**, permitiéndose que se vayan mejorando mientras el plazo de entrega esté abierto.

Las entregas se harán a través de la plataforma ATENEA.

Se irá publicando si las entregas mejoran la implementación de referencia y el ranking de las distintas entregas en cada categoría.