## FACULTAD DE INGENIERIA ELECTRICA Y COMPUTACION

## SISTEMAS OPERATIVOS

## MANUAL DE DESPLIEGUE

## SEGUNDO TERMINO 2015-2016

## INTEGRANTES:

CESAR SAN LUCAS

HERNAN ULLON

LUIS JACOME

# Sistema Operativo: UBUNTU 14.04

Para realizar la ejecución del programa en el Sistema Operativo Ubuntu o alguna otra distribución GNU/Linux, se debe tener instalado python2.7 (por defecto ya viene instalado), y se debe realizar el siguiente procedimiento para su ejecución.

# Procedimiento:

Para ejecutar el simulador de cache se debe descargar el proyecto desde el repositorio de Github disponible en <https://github.com/SLACE93/operativosFS>, luego descomprimirlo y dirigirse al directorio del proyecto mediante la terminal.

Por ejemplo si se encuentra la carpeta del proyecto en Downloads procedemos a ejecutar la siguiente línea comando

* cd Downloads/operativosFS

Una vez dentro del directorio del proyecto debemos generar el archivo workload.txt a través de las siguientes líneas de comando (es necesario tener conexión a internet) o si ha sido generado el archivo previamente copiarlo dentro del directorio del proyecto

wget <http://www.wikibench.eu/wiki/2007­10/wiki.1191201596.gz>

gunzip wiki.1191201596.gz

cut -f3 -d' ' wiki.1191201596 | grep [upload.wikimedia.org](http://upload.wikimedia.org/) | cut -c 28- > workload.txt

Luego através de la terminal ingresar la siguiente línea de comando

time python cacheSimulator.py <workload\_file> <policy> <cache\_size>

Para realizar la ejecución de la politica de reemplazo LRU con 1000 de cache size(entradas) debemos ingresar la siguiente línea de comando

time python cacheSimulator.py workload.txt LRU 1000

Una vez ejecutado nos mostrará los siguientes resultados:

Evaluando una cache LRU con 1000 entradas

Resultados:

Miss rate: 67.31% (2505271 misses out of 3721736 references)

Miss rate (warm cache): 67.31% (2504271 misses out of 3721736-1000 references)

real 1m23.209s

user 1m23.094s

sys 0m0.084s

Para realizar la ejecución de la politica de reemplazo CLOCK con 50000 de cache size(entradas) debemos ingresar la siguiente línea de comando

time python cacheSimulator.py workload.txt CLOCK 50000

Una vez ejecutado nos mostrará los siguientes resultados:

Evaluando una cache CLOCK con 50000 entradas

Resultados:

Miss rate: 41.14% (1531177 misses out of 3721736 references)

Miss rate (warm cache): 40.34% (1481177 misses out of 3721736-50000 references)

real 1m12.651s

user 1m12.571s

sys 0m0.108s

### **Bibliografía:**

* <http://en.wikipedia.org/wiki/Cache_algorithms>
* <http://en.wikipedia.org/wiki/Page_replacement_algorithm>
* http://algs4.cs.princeton.edu/34hash/