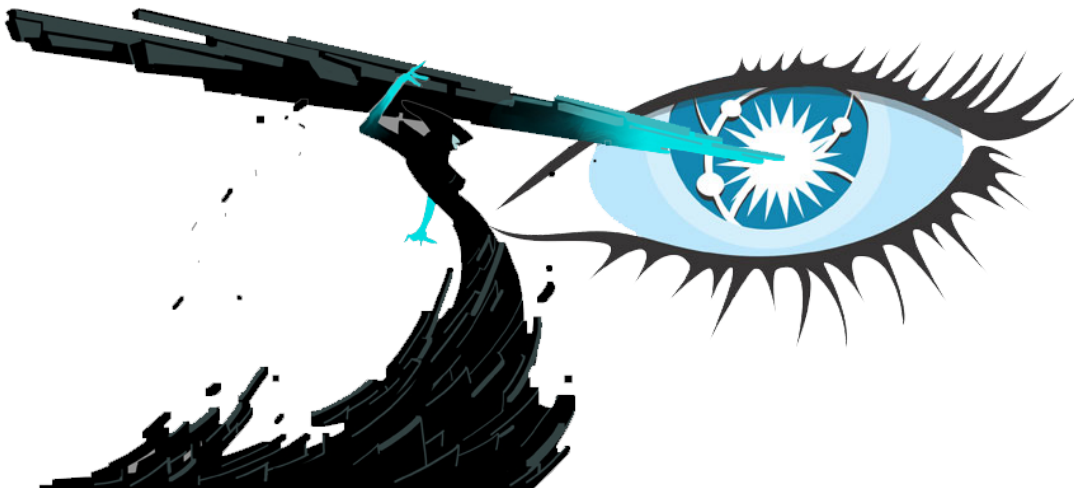


Ingeniería en Sistemas de Información

# Lissandra

## Documento de pruebas

*Cubriéndonos de aca a invierno*



Cátedra de Sistemas Operativos

Trabajo práctico Cuatrimestral

- 1C2019 -  
Versión 1.0

# Requisitos y notas de la evaluación

Los requisitos expuestos a continuación se encuentran ampliados en [las Normas del Trabajo Práctico](#), que por practicidad, se han resumido a continuación.

## Deploy y Setup

Es condición necesaria para la evaluación que **el Deploy y Setup del trabajo se realice en menos de 10 minutos**. Pasado este tiempo el grupo perderá el derecho a la evaluación.

Los archivos de configuración requeridos **para los diversos escenarios de pruebas** deberán ser preparados con anticipación por el grupo con todos los valores requeridos prefijados dejando sólo los parámetros desconocidos (ej: IP) incompletos.

## Compilación y ejecución

La compilación debe hacerse en la máquina virtual de la cátedra en su edición Server (no se pueden usar binarios subidos al repositorio).

Será responsabilidad del grupo verificar las dependencias requeridas para la compilación, y en caso de requerir bibliotecas provistas por la cátedra, descargarlas. También es responsabilidad de los integrantes del grupo conocer y manejar las herramientas de compilación desde la línea de comandos. Ver [Anexo - Comandos Útiles](#)

**< Comandos extras al compilar, como flags del sistema para sorting único u otros >**

# Prueba Mínima Kernel + Memoria

## Actividades:

Asociar la Memoria 1 al criterio SC y ejecutar el script `comidas.lql` y `animales.lql`. Luego de que comience a correr el script `comidas.lql`, ejecutar `misc_1.lql`. Una vez que finalice el script de `comidas.lql`, dentro de la consola de la Memoria ejecutar las siguientes request:

```
SELECT POSTRES 8927

INSERT FRUTAS 222 "Frutilla"

SELECT FRUTAS 621

INSERT POSTRES 12019 "Nutella"

INSERT FRUTAS 63 "Mango"

INSERT BEBIDAS 10 "Limonada"

SELECT FRUTAS 63
```

## Resultados Esperados:

- 1) Los tres scripts terminan correctamente.
- 2) Los scripts se planifican correctamente siguiendo un diagrama de estados y se respeta el Quantum entre los scripts de manera secuencial según Round Robin.
- 3) Las claves son correctamente actualizadas, finalizando de la siguiente forma:

Tabla	Clave	Valor
POSTRES	100	Helado
	80	Tiramisu
	101	Muffin
	4000	Gelatina
	8927	Helado
	231	Flan
	12019	Nutella
PLATOS_PRINCIPALES	271	Pizza
	32	Empanada
	54	Hamburguesa
BEBIDAS	10	Limonada
	11	Mate

	7338	Agua con gas
	44	Exprimido
FRUTAS	222	Frutilla
	621	Melon
	34	Cereza
	10245	Frutilla
	63	Mango
MARINOS	983	Pez globo
	33	Raya
	22	Delfin
	310	Calamar
	3103	Delfin
	3104	Delfin
	3105	Delfin
MAMIFEROS	2222	Elefante
	239	Burro
	21	Caballo
	349	Cebra
	721	Oveja
	272	Vaca
	209	Hipopotamo
AVES	11	Pinguino
	1102	Buho
	333	Colibri
	6262	Loro
	888	Pelicano
COLORES	9	Gris
	56278	Plateado
	8109	Verde
	11	Azul
	163	Rojo
	89	Verde

	3119	Verde
	00	Azul
COSAS	125	Celular
	124	Celular
	123	Celular
	332	Lapicera
	8291	Cable
	1	Libro
	889	Mouse
	38273	Computadora
ANIMALES	901	Paloma
	319	Oso hormiguero
	11	Oso
	318	Gato

### Configuración del sistema:

VM1 IP:      Proceso Kernel <htop> <consola limpia>	VM2 IP:      Proceso Memoria <htop> <consola limpia>
VM3 IP:      Proceso LFS <htop> <consola limpia>	VM4 IP:      -

### Archivos de Configuración

LFS            PUERTO\_ESCUCHA=5003  
                  PUNTO\_MONTAJE="/home/utnso/lissandra-checkpoint/"  
                  RETARDO=0  
                  TAMAÑO\_VALUE=15  
                  TIEMPO\_DUMP=60000

Memoria      PUERTO=8001  
                  PUERTO\_FS=5003  
                  IP\_SEEDS=[ ]  
                  PUERTO\_SEEDS=[ ]  
                  RETARDO\_MEM=600  
                  RETARDO\_FS=600  
                  TAM\_MEM=4096

RETARDO\_JOURNAL=70000  
RETARDO\_GOSSIPING=30000  
MEMORY\_NUMBER=1

Kernel      PUERTO\_MEMORIA=8001  
              QUANTUM=3  
              MULTIPROCESAMIENTO=1  
              METADATA\_REFRESH=15000  
              SLEEP\_EJECUCION=100

Metadata FileSystem    BLOCK\_SIZE=128  
                          BLOCKS=4096  
                          MAGIC\_NUMBER=LISSANDRA

# Prueba Mínima LFS

## Disclaimer

El objetivo de esta prueba es realizar y verificar los resultados en base hasta lo que el grupo tenga implementado. Los resultados ofrecidos por esta prueba contienen los valores esperados para un LFS desarrollado en un 100% (memtable + dump + temporales + compactación).

La verificación a realizar por cada grupo consta en realizar operaciones GET sobre las keys que figuren en las particiones obteniendo su valor final (independiente del nivel de desarrollo que posee al LFS al momento de la evaluación).

## Actividades:

Abrir una consola de LFS y ejecutar secuencialmente las sentencias que se encuentran en el script `películas.1ql` con los siguientes valores en donde figura [\[particiones\]](#):

- a) 5
- b) 7
- c) 2

## Resultados Esperados:

Las claves son correctamente particionadas de la siguiente forma:

a)

Partición 0	Partición 1	Partición 2	Partición 3	Partición 4
10;Toy Story 1110;Harry Potter 13535;Titanic	2516;Godzilla 3671;Avatar	922;Ratatouille	163;Nemo	4829;Aladdin

b)

P0	P1	P2	P3	P4	P5	P6
		163;Nemo	10;Toy Story 2516;Godzilla 3671;Avatar	1110;Harry Potter 13535;Titanic	922;Ratatouille	4829;Aladdin

c)

Partición 0	Partición 1
10;Toy Story 1110;Harry Potter 922;Ratatouille 2516;Godzilla	163;Nemo 13535;Titanic 4829;Aladdin 3671;Avatar

## Configuración del sistema:

VM1 IP:	Proceso LFS	VM2 IP:
---------	-------------	---------

<htop> <consola limpia>	
VM3 IP:	VM4 IP: -

#### Archivos de Configuración

LFS            PUERTO\_ESCUCHA=5005  
                  PUNTO\_MONTAJE="/home/utnso/lissandra-checkpoint/"  
                  RETARDO=100  
                  TAMAÑO\_VALUE=255  
                  TIEMPO\_DUMP=60000

Metadata FileSystem   BLOCK\_SIZE=64  
                          BLOCKS=5192  
                          MAGIC\_NUMBER=LISSANDRA



# Prueba Error Kernel + Memoria

## Actividades:

Asociar la Memoria 3 al criterio SC y ejecutar el script `misc_1.lql` y `misc_2.lql`. Luego de que comience a correr el script `misc_1.lql`, ejecutar `animales_falla.lql`. Una vez que falle `animales_falla.lql` y finalicen los scripts anteriores lanzar `comidas.lql` y luego `misc_1_falla.lql`

## Resultados Esperados:

- 1) `misc_1.lql`, `misc_2.lql` y `comidas.lql` finalizan correctamente
- 2) `animales_falla.lql` y `misc_1_falla.lql` abortan su ejecución
- 3) Los scripts se planifican correctamente siguiendo un diagrama de estados y se respeta el Quantum entre los scripts de manera secuencial según Round Robin.
- 4) Las claves son correctamente actualizadas, finalizando de la siguiente forma:

COLORES	9	Gris
	56278	Plateado
	8109	Verde
	11	Azul
	163	Rojo
	89	Verde
	3119	Amarillo
	00	Azul
COSAS	125	Celular
	124	Celular
	123	Celular
	332	Lapicera
	8291	Cable
	1	Libro
	889	Mouse
	38273	Computadora
ANIMALES	901	Paloma
	319	Oso hormiguero
	11	Oso
	318	Gato

PELICULAS	124	UP
	3128	Big fish
	332	Lapicera
	123	Taza
	56278	UP
	38273	Monsters inc
	10	Nemo
	9	Shrek
SERIES	163	Friends
	332	Lie to me
	89	Lost
LIBROS	42229	Heidi
	18902	El principito
	901	El hobbit
	319	Crepusculo
	190	Dracula
MARINOS	983	Ballena
POSTRES	100	Helado
	80	Tiramisu
	101	Muffin
	4000	Gelatina
	8927	Helado
	231	Flan
	12019	Cheesecake
PLATOS_PRINCIPALES	271	Pizza
	32	Empanada
	54	Hamburguesa
BEBIDAS	10	Limonada
	11	Mate
	7338	Agua con gas
	44	Exprimido
FRUTAS	222	Frutilla

	621	Melon
	34	Cereza
	10245	Frutilla
NOVELAS_ARGENTINAS	9	Patito feo
	10	Floricienta

### Configuración del sistema:

VM1 IP:	Proceso Kernel <htop> <consola limpia>	VM2 IP:	Proceso Memoria <htop> <consola limpia>
VM3 IP:	Proceso LFS <htop> <consola limpia>	VM4 IP:	-

### Archivos de Configuración

LFS            PUERTO\_ESCUCHA=5005  
                  PUNTO\_MONTAJE="/home/utnso/lissandra-checkpoint/"  
                  RETARDO=0  
                  TAMAÑO\_VALUE=15  
                  TIEMPO\_DUMP=60000

Memoria       PUERTO=8006  
                  PUERTO\_FS=5005  
                  IP\_SEEDS=[]  
                  PUERTO\_SEEDS=[]  
                  RETARDO\_MEM=600  
                  RETARDO\_FS=600  
                  TAM\_MEM=4096  
                  RETARDO\_JOURNAL=70000  
                  RETARDO\_GOSSIPING=30000  
                  MEMORY\_NUMBER=3

Kernel        PUERTO\_MEMORIA=8006  
                  QUANTUM=3  
                  MULTIPROCESAMIENTO=1  
                  METADATA\_REFRESH=15000  
                  SLEEP\_EJECUCION=100

Metadata FileSystem `BLOCK_SIZE=32`  
`BLOCKS=2048`  
`MAGIC_NUMBER=LISSANDRA`

## Anexo - Comandos Útiles

### Copiar un directorio completo por red

```
scp -rpC [directorio] [ip]:[directorio]
```

Ejemplo:

```
scp -rpC tp-1c2015-repo 192.168.3.129:/home/utnso
```

### Descargar bibliotecas en un repositorio (como las commons)

```
git clone [url_repo]
```

Ejemplo:

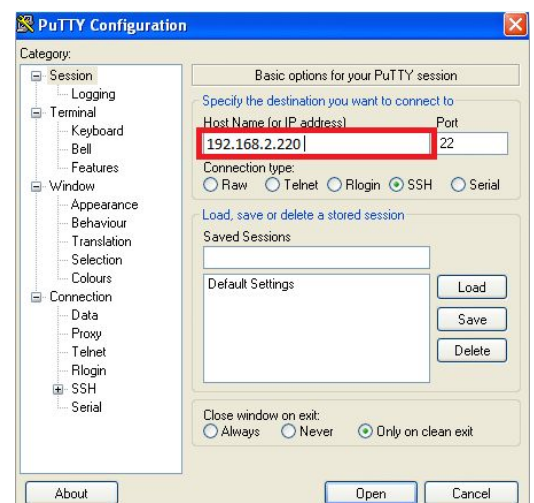
```
git clone https://github.com/sisoputnfrba/so-commons-library
```

## PuTTY

Este famoso utilitario nos permite desde Windows acceder de manera simultánea a varias terminales de la Máquina Virtual, similar a abrir varias terminales en el entorno gráfico de Ubuntu.

Ya se encuentra en las computadoras del laboratorio y se puede descargar desde [aquí](#)

Al iniciar debemos ingresar la IP de nuestra máquina virtual en el campo **Host Name (or IP address)** y luego presionar el botón **Open** y loguearnos como **utnso**



### Se recomienda investigar:

- Directorios y archivos: cd, ls, mv, rm, ln (creación de symlinks)
- Entorno: export, variable de entorno LD\_LIBRARY\_PATH
- Compilación: make, gcc, makefile
- Criptografía: md5sum
- Visor de procesos del sistema: htop.