

Ingeniería en Sistemas de Información

# Cache 13

## Documento de pruebas



Cátedra de Sistemas Operativos

Trabajo práctico Cuatrimestral

- 2C2015 -  
Versión [0.2.2]

# Requisitos y notas de la evaluación

## Deploy y Setup

- Es condición necesaria para la evaluación que **el Deploy & 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 por el grupo con anticipación dejando sólo los parámetros desconocidos (ej: IP) incompletos.
- En la fecha de entrega la conexión a Internet podría estar congestionada para clonar el repositorio desde GitHub. Debido a eso **el grupo debe traer una copia del trabajo en un medio extraíble**, subirlo a una máquina virtual y luego copiar dicho repositorio por red entre las VMs. Ver [Anexo - Comandos Útiles](#)

## 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).
- Para facilitar la visualización de varias terminales de manera simultánea se utilizará la herramienta **PuTTY** para acceder a las consolas de las Máquinas Virtuales.
- Es responsabilidad del grupo verificar que los parámetros de compilación sean portables y conocer y manejar las herramientas de compilación desde la línea de comandos. Ver [Anexo - Comandos Útiles](#)
- Debido a la complejidad y la concurrencia de los eventos que se van a evaluar es imprescindible que el alumno verifique que **su registro (log) permita determinar en todo momento el estado actual y anterior del sistema** y sus cambios significativos.

## Evaluación Final

- Cada grupo deberá llevar **dos** copias impresas de la [planilla de evaluación](#)<sup>1</sup> con los datos de los integrantes completos (dejando los campos “Nota” y “Coloquio” en blanco) y una copia de los presentes tests.
- Las pruebas pueden ser alteradas o modificadas entre instancias de entrega y recuperatorios, y podrán ser adaptadas durante el transcurso de la corrección a criterio del ayudante para lograr validar el correcto funcionamiento y desempeño del sistema desarrollado.
- En los casos en que las modificaciones se vuelvan permanentes, el documento será actualizado y re-publicado para reflejar estos cambios.

---

<sup>1</sup> Al final de este documento



# Pruebas - Checkpoint 2

Las siguientes pruebas son orientativas. Cada evaluador realizará las modificaciones que considere necesarias, para probar los Trabajos Prácticos.

## Configuración inicial

### Planificador

PUERTO\_ESCUCHA: 4000  
ALGORITMO\_PLANIFICACION: FIFO

### Administrador de Memoria

PUERTO\_ESCUCHA: 5000  
IP\_SWAP: 127.0.0.1  
PUERTO\_SWAP: 6000  
MAXIMO\_MARCOS\_POR\_PROCESO: 0  
CANTIDAD\_MARCOS: 0  
TAMANIO\_MARCO : 0  
ENTRADAS\_TLB: 0  
TLB\_HABILITADA: NO  
RETARDO\_MEMORIA: 10

### CPU

IP\_PLANIFICADOR: 127.0.0.1  
PUERTO\_PLANIFICADOR: 4000  
IP\_MEMORIA: 127.0.0.1  
PUERTO\_MEMORIA: 5000  
CANTIDAD\_HILOS: 1  
RETARDO: 1

### Administrador de Swap

PUERTO\_ESCUCHA: 6000  
NOMBRE\_SWAP: swap.data  
CANTIDAD\_PAGINAS: 64  
TAMANIO\_PAGINA: 4  
RETARDO\_SWAP: 5  
RETARDO\_COMPACTACION: 60

## Ejemplos de mCod

largo.cod	corto.cod	no_entra.cod	entra.cod
<pre>iniciar 10 leer 0 leer 1 leer 2 leer 3 leer 4 leer 9 leer 8 leer 7 leer 6 leer 5 finalizar</pre>	<pre>iniciar 2 leer 0 leer 1 finalizar</pre>	<pre>iniciar 100 leer 0 finalizar</pre>	<pre>iniciar 64 leer 0 leer 1 leer 2 leer 60 leer 63 finalizar</pre>

## Prueba 1 - Condición mínima

Esta prueba comprueba el estado determinado como mínimo para que un trabajo práctico sea evaluado. Si esto no ocurre, de todas formas es importante que el grupo asista para evacuar sus dudas. El checkpoint es una instancia de aprendizaje y feedback.

Se deberá correr `largo.cod` correctamente. Al revisar los logs, se mostrarán las sentencias ejecutadas.

## Prueba 2 - Ejecución FIFO

Se deberá correr de forma concurrente el código de `largo.cod`, `corto.cod` y `largo.cod` (nuevamente). Al revisar los logs, se mostrarán las sentencias ejecutadas, en orden FIFO.

## Prueba 3 - Rechazo de proceso

Se deberá correr de forma concurrente el código de `largo.cod` y `no_entra.cod`. `no_entra.cod` deberá ser rechazado.

## Prueba 4 - Caso borde

Se deberá correr el código de `entra.cod`. La partición de swap de 64 posiciones debería estar completamente ocupada.

## Prueba 5 - Opcional

En el código que crea la partición de swap, se cambiará el caracter `'0'` con el que se inicializa, por una `A`. Correr las pruebas 1 a 4, chequeando que todas las lecturas devuelvan `"AAAA"`.

# Planilla de Evaluación - TP2C2015

Grupo:

Legajo	Nombre y Apellido	Nota

Evaluador:

Coloquio:

---

Condiciones Mínimas	
A definir	

Planificador	

CPU	

Administrador de Memoria	

Administrador de Swap	

# 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 **solo** la última versión del código (en vez de todo el repositorio)

```
curl -u '[usuario]' -L -o [archivo] [url_repo]
```

Ejemplo:

```
curl -u 'gastonprieto' -L -o commons.tar
```

<https://api.github.com/repos/sisoputnfrba/so-commons-library/tarball/master>

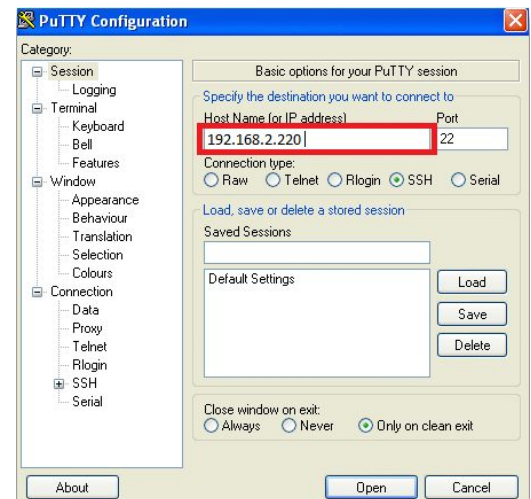
Este comando debe ejecutarse sin salto de línea. Luego **descomprimir** con: `tar -xvf commons.tar`

## 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`