

Ingeniería en Sistemas de Información

[Elestac]

el producto
altera
el orden de los factores



Documento de pruebas

Cátedra de Sistemas Operativos

Trabajo práctico Cuatrimestral

- 1C2016 -
Versión [0.1]

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.

Los scripts ansisop que se piden se encuentran en el siguiente repositorio: <https://github.com/sisoputnfrba/scripts-ansisop>

En la fecha de entrega la conexión a Internet podría estar congestionada para clonar el repositorio desde GitHub. Debido a eso *se recomienda traer una copia del trabajo en un medio extraíble* e investigar métodos para copiar directorios entre máquinas en red (scp/WinSCP).

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). 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](#)

Evaluación

Cada grupo deberá llevar **dos** copias impresas de la [planilla de evaluación](#)¹ con los datos de los integrantes completos (dejar el campo “Nota” y “Coloquio” en blanco) y una copia de los presentes tests.

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.

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. En otras palabras, este documento es de carácter orientativo: el ayudante podrá realizar o ignorar las pruebas que considere apropiadas para lograr el objetivo de la corrección - verificar 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.

¹ Al final de este documento

Pruebas

Prueba 1 - Prueba Base

Mediante este test se validará un ciclo básico del sistema.

Configuración

VM1 Núcleo Programa 1 - facil.ansisop	VM3 CPU 2 Swap Programa 2 - consumidor.ansisop
VM2 CPU 1 Programa 3 - productor.ansisop	VM4 UMC Programa 4 - consumidor.ansisop

Configuración del Núcleo:

```
QUANTUM=3
QUANTUM_SLEEP=1000
SEM_IDS=[b, c]
SEM_INIT=[1, 0]
IO_IDS=[HDD1, LPT1]
IO_SLEEP=[500, 200]
SHARED_VARS=[colas,compartida]
STACK_SIZE=8
```

Configuración del Swap:

```
CANTIDAD_PAGINAS=256
TAMANIO_PAGINA2=64
RETARDO_ACCESO=3000
RETARDO_COMPACTACION=10000
```

Configuración de la UMC:

```
MARCOS=5
MARCO_SIZE=64
MARCO_X_PROC=2
ALGORITMO=CLOCK
ENTRADAS_TLB=3
RETARDO=2500
```

Desarrollo

Iniciar el Núcleo, la UMC, el Swap y las CPUs 1 y 2. Validar las conexiones, uso de memoria, cantidades de hilos y CPU de los procesos. Ejecutar el script facil.ansisop en la VM1 (Programa 1), observando el tiempo de espera entre quantums. Verificar en la UMC que se creen las estructuras, almacenando el código solamente en Swap. Esperar a que finalice el script, y validar que la UMC y el swap hayan quedado limpios.

Ejecutar el script consumidor.ansisop en la VM2 (Programa 2). Verificar que el Programa se bloquea esperando por el semáforo c.

Ejecutar el script productor.ansisop en la VM 3 (Programa 3). Verificar que los Programas alternan su ejecución y bloqueo, y que consumidor muestra números consecutivos.

Ejecutar el script consumidor.ansisop en la VM 4 (Programa 4). Verificar que los Programas respetan el quantum y que los scripts son planificados según Round Robin.

² Si aplicase

Prueba 2 - Prueba Integral

Mediante este test se validará el funcionamiento íntegro del sistema, utilizando todas las primitivas provistas por la especificación AnSISOP, verificando además el correcto uso de memoria y CPU por parte de los procesos.

Configuración

VM1 Núcleo Programa 1 - completo.ansisop	VM3 CPU 2 Swap
VM2 CPU 1 Programa 2 - productor.ansisop	VM4 UMC CPU3 Programa 3 - consumidor.ansisop

Configuración del Núcleo:

```
QUANTUM=3
QUANTUM_SLEEP=250
SEM_IDS=[b, c]
SEM_INIT=[1, 0]
IO_IDS=[HDD1, LPT1]
IO_SLEEP=[500, 200]
SHARED_VARS=[colas, compartida]
STACK_SIZE=8
```

Configuración del Swap:

```
CANTIDAD_PAGINAS=256
TAMANIO_PAGINA3=64
RETARDO_ACCESO=1500
RETARDO_COMPACTACION=3000
```

Configuración de la UMC:

```
MARCOS=5
MARCO_SIZE=64
MARCO_X_PROC=2
ALGORITMO=CLOCK
ENTRADAS_TLB=3
RETARDO=1000
```

Desarrollo

Iniciar el Núcleo, la UMC, el Swap y las CPUs 1 y 2. Ejecutar los scripts productor.ansisop (VM2 Programa 2) y consumidor.ansisop (VM Programa 3).

Luego de un par de iteraciones, ejecutar el script completo.ansisop en la VM1 (Programa 1), verificar el reemplazo de marcos en memoria y la utilización de la TLB. Ejecutar la CPU3, y verificar que los Programas comienzan a ejecutarse en ella.

Verificar que el Programa 5 concluye su ejecución, y luego enviar la señal SIGUSR1 una CPU que se encuentre ejecutando. Validar que se cierre al finalizar la ráfaga que está ejecutando, y que todos Programas continúan su ejecución. Reconectar la CPU y validar que se efectivamente se encuentre conectada. Volver a realizar este procedimiento, pero desconectando la CPU de forma abortiva y verificar que el programa se finaliza y el sistema sigue funcionando.

³ Si aplicase

Prueba 3 - Prueba Planificación

Mediante este test se validará el funcionamiento de entrada y salida, planificador y estrés. Como antes, se verificará los consumos de memoria y CPU.

Configuración

VM1 Núcleo Programa 1 - forES.ansisop	VM3 Programa 3: pesado.ansisop Swap
VM2 CPU 1 Programa 2 - forES.ansisop	VM4 UMC Programa 4 - fibo.ansisop

Configuración del Núcleo:

QUANTUM=3
QUANTUM_SLEEP=1000
SEM_IDS=[b, c]
SEM_INIT=[1, 0]
IO_IDS=[HDD1, LPT1]
IO_SLEEP=[500, 200]
SHARED_VARS=[colas, compartida]
STACK_SIZE=8

Configuración del Swap:

CANTIDAD_PAGINAS=256
TAMANIO_PAGINA=64
RETARDO_ACCESO=3000
RETARDO_COMPACTACION=10000

Configuración de la UMC:

MARCOS=5
MARCO_SIZE=64
MARCO_X_PROC=2
ALGORITMO=CLOCK
ENTRADAS_TLB=0
RETARDO=2500

Desarrollo

Se ejecutarán en orden los programas 1, 2, 3 y 4. Durante la ejecución, disminuir el valor del quantum y el tiempo del mismo y observar los cambios.

Una vez finalizado el programa 3, desconectar un proceso consola asociado a alguno de los programas de forma abortiva.

Resultados Esperados

Se deberá respetar el Round Robin según orden: 1, 2, 3 (finalizando tras su ejecución) y 4 al principio y luego se irá alterando de acuerdo a los tiempos de E/S.

Se observará que el tiempo de Quantum impacta en la próxima ejecución de un programa en la CPU.

Al desconectarse un proceso consola, el sistema debe seguir funcionando, debiendo eliminar el PCB y liberar la memoria asignada.

Prueba 4 - Prueba Memoria

Mediante este test se validará el funcionamiento del módulo de memoria y de Swap. Se evaluarán los algoritmos de de reemplazo de TLB y marcos en memoria, limpieza de registros, integridad de los pedidos y el algoritmo de compactación del Swap.

Configuración

VM1 Núcleo Programa 1 - vector.ansisop	VM3 Programa 4: stackoverflow.ansisop Swap CPU 3
VM2 CPU 1 Programa 2 - vector.ansisop	VM4 UMC Programa 3 - vector.ansisop

Configuración del Núcleo:

```
QUANTUM=5
QUANTUM_SLEEP=500
SEM_IDS=[b, c]
SEM_INIT=[1, 0]
IO_IDS=[HDD1, LPT1]
IO_SLEEP=[500, 200]
SHARED_VARS=[colas, compartida]
STACK_SIZE=16
```

Configuración del Swap:

```
CANTIDAD_PAGINAS=256
TAMANIO_PAGINA=32
RETARDO_ACCESO=4000
RETARDO_COMPACTACION=10000
```

Configuración de la UMC:

```
MARCOS=4
MARCO_SIZE=32
MARCO_X_PROC=3
ALGORITMO=CLOCK-M
ENTRADAS_TLB=3
RETARDO=2500
```

Desarrollo

Se ejecutarán los programas 1 y 3, chequeando el funcionamiento de la TLB y del algoritmo LRU de el funcionamiento de la misma, así como el algoritmo Clock-M de la UMC. Ingresar el programa 2 una vez ocupados los 4 Marcos y verificar que eventualmente se finalice dicho proceso.

Realizar un Flush de la TLB y verificar los TLB miss. Realizar un Dump de la memoria. Luego efectuar un Flush de Memoria, verificar las escrituras a Swap y el estado de la memoria utilizando el comando Dump nuevamente.

Tras finalizar la ejecución de todos los programas, ejecutar los programas 1 y 2. Finalizar la consola del programa 2 manualmente. Volver a ejecutarlo y compactar el Swap. Finalmente, reducir el tamaño del Swap a 100 y volver a ejecutar los programas 1 a 3 y verificar que un script no se inicia por no poder reservar suficiente espacio en Swap. Ejecutar el programa 4 y verificar el desbordamiento del stack.

Resultados Esperados

A definir

Planilla de Evaluación - TP1C2016

Grupo:

Legajo	Nombre y Apellido	Nota

Evaluador:

Coloquio:

Prueba 1 - Prueba Base	
Las CPUs ejecutan de forma simultánea e independiente	
El código se almacena en el Swap inicialmente y se carga a demanda en la UMC	
La planificación de RR se respeta	
Al terminar un programa, la UMC y el Swap quedan limpios	
Se respeta el correcto funcionamiento de System Calls (semáforos y variables compartidas)	
El uso de CPU y memoria no es excesivo (top). La sincronización no depende de sleep()	
La cantidad de hilos en el sistema es la adecuada	
El estado de las colas de estado (Listos, Bloqueados) del Kernel es válido y comprensible.	

Prueba 2: Prueba Integral

Los CPUs pueden ingresar y salir del sistema en cualquier momento tanto de forma abortiva como de forma segura

Al enviar SIGUSR1 al CPU, éste se cierra al concluir la ráfaga de ejecución actual, y el Programa puede seguir ejecutando en otras CPUs

La ejecución del script **completo** es independiente de los otros programas

No hay esperas activas

Prueba 3: Prueba Planificación

El algoritmos de Clock se respeta

El tiempo de entrada y salida se respeta y se atiende a los pedidos de forma correcta

El Quantum es configurable y se respeta

El tiempo de espera de Quantums es configurable y se respeta

La desconexión de un proceso consola no afecta al funcionamiento del sistema

Prueba 4: Prueba Memoria

Se respeta el correcto funcionamiento de la memoria. UMV -> TLB -> Tabla de Páginas -> SWAP

Un programa que no tiene frames disponibles al demandarlo es terminado

Un programa que no puede reservar páginas suficientes en Swap es terminado

La UMV puede atender, en simultáneo, pedidos del sistema y del usuario

El algoritmo Clock-M se respeta

El algoritmo de compactación es correcto

El retardo se respeta y es configurable desde la consola

La tabla de páginas es legible y correcta

La TLB es utilizada correctamente y se respeta el algoritmo LRU

La limpieza de la TLB se realiza efectivamente

La limpieza de la UMC impacta sobre el Swap

Los errores en los procesos no impiden el funcionamiento del resto del sistema
(**stackoverflow.anisop**)

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 [nombre_del_archivo_a_generar] [url_repo]
```

Ejemplo:

```
curl -u 'gastonprieto' -L -o commons_ultimo.tar  
https://api.github.com/repos/sisoputnfrba/so-commons-library/tarball/master
```

*Este comando debe ejecutarse sin salto de línea.

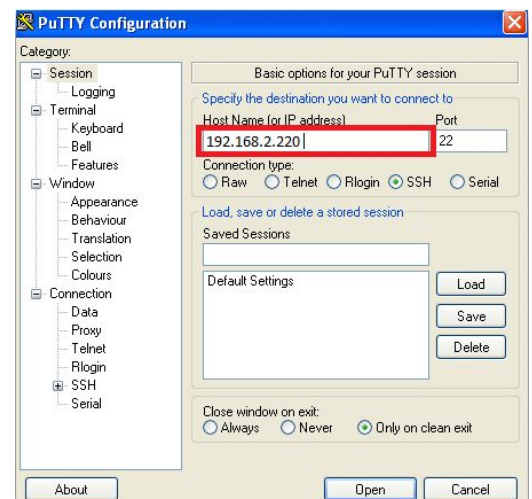
Luego descomprimir con: `tar -xvf commons_ultimo.tar`

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