

Ingeniería en Sistemas de Información

GECK

Documento de pruebas

Tests... Tests never changes



Cátedra de Sistemas Operativos

Trabajo práctico Cuatrimestral

-2C2022 -
Versión 2.1

Índice

Índice	2
Versión de Cambios	3
Aclaraciones	4
Prueba Base	5
Actividades	5
Resultados Esperados	5
Configuración del sistema	5
Prueba Planificación Base	7
Actividades	7
Resultados Esperados	7
Configuración del sistema	7
Prueba Planificación Completa	9
Actividades	9
Resultados Esperados	9
Configuración del sistema	9
Prueba Memoria	11
Actividades	11
Resultados Esperados	11
Configuración del sistema	11
Prueba TLB	13
Actividades	13
Resultados Esperados	13
Configuración del sistema	13
Prueba Estabilidad General	15
Actividades	15
Resultados Esperados	15
Configuración del sistema	15

Versión de Cambios

v1.0 (22/10/2022) Publicación Inicial Check Obligatorio

v2.0 (15/11/2022) Publicación de pruebas para las Entregas Finales

v2.1 (29/11/2022) Ajustes en configs

Aclaraciones

Dadas las condiciones del trabajo práctico, el objetivo de este documento es orientar a los alumnos a pruebas que permitirán la evaluación del trabajo práctico.

Todos los scripts para realizar las pruebas que se enumeran en este documento se encuentran subidos al repositorio: <https://github.com/sisoputnfrba/geck-pruebas>

Dentro de las configuraciones propuestas en cada prueba puede haber casos de algunos procesos que no tengan su respectiva configuración porque son valores que no afectan a la prueba en sí.

Los datos de los config que no son provistos en el documento de pruebas es porque dependen de la distribución realizada por los alumnos (por ejemplo IPs, Puertos o Paths).

Será responsabilidad del grupo verificar las dependencias requeridas para la compilación, y en caso de requerir bibliotecas provistas por la cátedra, descargarlas.

Por último y no menos importante, está totalmente prohibido subir archivos binarios al repositorio.

Prueba Base

Actividades

1. Iniciar los módulos
2. Ejecutar 3 módulos consola con los siguientes scripts en el orden indicado:
 - a. BASE_1
 - b. BASE_2
 - c. BASE_3
3. Esperar su finalización
4. Cambiar el Algoritmo a RR y esperar la finalización
5. Cambiar el algoritmo a FEEDBACK y esperar la finalización

Resultados Esperados

- Con el algoritmo FIFO los procesos finalizan en el orden 1-2-3.
- Con el algoritmo RR los procesos finalizan en el orden 2-1-3.
- Con el algoritmo FEEDBACK finalizan en el orden 1-2-3 pero ejecutan los 3 de manera intercalada (simil RR) antes de que finalice el 1.

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_CPU=127.0.0.1 PUERTO_CPU_DISPATCH=8001 PUERTO_CPU_INTERRUPT=8005 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO GRADO_MAX_MULTIPROGRAMACION=4 DISPOSITIVOS_IO=[DISCO, IMPRESORA] TIEMPOS_IO=[1000, 2000] QUANTUM_RR=1500	ENTRADAS_TLB=4 REEMPLAZO_TLB=LRU RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA_DISPATCH=8001 PUERTO_ESCUCHA_INTERRUPT=8005
<i>Memoria.config</i>	
PUERTO_ESCUCHA=8002 TAM_MEMORIA=4096 TAM_PAGINA=64 ENTRADAS_POR_TABLA=4 RETARDO_MEMORIA=1000 ALGORITMO_REEMPLAZO=CLOCK MARCOS_POR_PROCESO=4 RETARDO_SWAP=2000	

```
PATH_SWAP=/home/utnso/swap  
TAMANIO_SWAP=10240
```

Prueba Planificación Base

Actividades

1. Iniciar los módulos
2. Ejecutar 4 módulos consola con los siguientes scripts en el orden indicado:
 - a. PLANI_BASE_1
 - b. PLANI_BASE_2
 - c. PLANI_BASE_3
 - d. PLANI_BASE_4
3. Esperar la finalización de los programa 2, 3 y 4 y recién ahí ingresar un valor por teclado para finalizar 1
4. Cambiar de algoritmo a RR y repetir los pasos 1, 2 y 3.
5. Cambiar de algoritmo a FEEDBACK y repetir los pasos 1, 2 y 3.

Resultados Esperados

- Con FIFO los procesos terminan en el siguiente orden: 2-3-4-1
- Con RR los proceso terminan en el siguiente orden 2-4-3-1
- Con FEEDBACK los procesos terminan en el orden 2-4-3-1 pero tanto P2 como P3 pasan por la cola de FIFO ya que son desalojados por fin de quantum

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_CPU=127.0.0.1 PUERTO_CPU_DISPATCH=8001 PUERTO_CPU_INTERRUPT=8005 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO GRADO_MAX_MULTIPROGRAMACION=4 DISPOSITIVOS_IO=[DISCO, IMPRESORA, WIFI, AUDIO, USB] TIEMPOS_IO=[1000, 2000, 5000, 500, 4000] QUANTUM_RR=1500	ENTRADAS_TLB=4 REEMPLAZO_TLB=LRU RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA_DISPATCH=8001 PUERTO_ESCUCHA_INTERRUPT=8005
<i>Memoria.config</i>	<i>Consola.config</i>
PUERTO_ESCUCHA=8002 TAM_MEMORIA=4096 TAM_PAGINA=64 ENTRADAS_POR_TABLA=4	IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[16, 16, 16, 16] TIEMPO_PANTALLA=750

RETARDO_MEMORIA=1000 ALGORITMO_REEMPLAZO=CLOCK MARCOS_POR_PROCESO=4 RETARDO_SWAP=2000 PATH_SWAP=/home/utnso/swap TAMANIO_SWAP=10240	
--	--

Prueba Planificación Completa

Actividades

1. Iniciar los módulos
2. Ejecutar 4 módulos consola con los siguientes scripts en el orden indicado:
 - a. PLANI_FULL_1
 - b. PLANI_FULL_1
 - c. PLANI_FULL_2
 - d. PLANI_FULL_2
3. Esperar que los procesos A y B queden frenados por entrada de teclado.
4. Desbloquear los procesos A y B ingresando cualquier valor por teclado y esperar su finalización.
5. Esperar la finalización del resto de los procesos.

Resultados Esperados

- Al momento de que los procesos A y B se bloquean por teclado no ingresan nuevos procesos a la cola de READY.
- Los procesos A y B se van alternando en su ejecución por ser I/O Bound, ejecutando en ciertos momentos I/O en paralelo en diferentes dispositivos.
- Los procesos C y D con FIFO ejecutan en el orden que se desbloquean los teclados mientras que con RR se van alternando ya que son CPU Bound.
- Los procesos C y D en el algoritmo Feedback ejecutan una rafaga en RR y después cuando vuelven a ingresar ejecutan sin detenerse hasta el fin por ser CPU Bound.

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_CPU=127.0.0.1 PUERTO_CPU_DISPATCH=8001 PUERTO_CPU_INTERRUPT=8005 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO GRADO_MAX_MULTIPROGRAMACION=2 DISPOSITIVOS_IO=[DISCO, IMPRESORA] TIEMPOS_IO=[4000, 8000] QUANTUM_RR=2500	ENTRADAS_TLB=4 REEMPLAZO_TLB=LRU RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA_DISPATCH=8001 PUERTO_ESCUCHA_INTERRUPT=8005
<i>Memoria.config</i>	<i>Consola.config</i>
PUERTO_ESCUCHA=8002 TAM_MEMORIA=4096	IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000

TAM_PAGINA=64 ENTRADAS_POR_TABLA=4 RETARDO_MEMORIA=1000 ALGORITMO_REEMPLAZO=CLOCK MARCOS_POR_PROCESO=4 RETARDO_SWAP=2000 PATH_SWAP=/home/utnso/swap TAMANIO_SWAP=10240	SEGMENTOS=[16, 16, 16, 16] TIEMPO_PANTALLA=750
---	---

Prueba Memoria

Actividades

1. Iniciar los módulos.
2. Ejecutar 3 módulos consola con los siguientes scripts en el orden indicado:
 - a. MEMORIA_1
 - b. MEMORIA_2
 - c. MEMORIA_3
3. Esperar que los procesos finalicen.

Resultados Esperados

- El resultado impreso por pantalla por los procesos A y B es el mismo.
- El proceso C finaliza con un *Segment Fault*.
- El proceso A se bloquea 6 veces por page fault.
- El proceso B sólo deberá bloquearse 1 vez al cargar la primera página.
- La cantidad de accesos a SWAP del proceso A difieren entre Clock y Clock-M.

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_CPU=127.0.0.1 PUERTO_CPU_DISPATCH=8001 PUERTO_CPU_INTERRUPT=8005 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO GRADO_MAX_MULTIPROGRAMACION=1 DISPOSITIVOS_IO=[DISCO, IMPRESORA] TIEMPOS_IO=[1000, 5000] QUANTUM_RR=2500	ENTRADAS_TLB=0 REEMPLAZO_TLB=LRU RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA_DISPATCH=8001 PUERTO_ESCUCHA_INTERRUPT=8005
<i>Memoria.config</i>	<i>Consola.config (A)</i>
PUERTO_ESCUCHA=8002 TAM_MEMORIA=4096 TAM_PAGINA=64 ENTRADAS_POR_TABLA=2 RETARDO_MEMORIA=1000 ALGORITMO_REEMPLAZO=CLOCK/CLOCK-M MARCOS_POR_PROCESO=2 RETARDO_SWAP=5000 PATH_SWAP=/home/utnso/swap	IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[64, 96, 128, 32] TIEMPO_PANTALLA=750

TAMANIO_SWAP=10240	
<i>Consola.config (B)</i>	<i>Consola.config (C)</i>
IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[64, 64, 96, 128] TIEMPO_PANTALLA=750	IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[64] TIEMPO_PANTALLA=750

Prueba TLB

Actividades

1. Iniciar los módulos
2. Ejecutar 2 módulos consola con los siguientes scripts en el orden indicado:
 - a. TLB_1
 - b. TLB_2
3. Esperar que el proceso A imprima por pantalla e ingresar por el teclado del proceso B el valor 228.
4. Esperar que el proceso B finalice e ingresar cualquier valor en el teclado del proceso A y esperar su finalización.

Resultados Esperados

- Los Hits de la TLB en el proceso A difieren según el algoritmo.
- El proceso B reemplaza diferentes entradas de la TLB según el algoritmo.
- El proceso B finaliza por *Segment Fault*.

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_CPU=127.0.0.1 PUERTO_CPU_DISPATCH=8001 PUERTO_CPU_INTERRUPT=8005 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO GRADO_MAX_MULTIPROGRAMACION=2 DISPOSITIVOS_IO=[DISCO, IMPRESORA] TIEMPOS_IO=[1000, 5000] QUANTUM_RR=2500	ENTRADAS_TLB=4 REEMPLAZO_TLB=FIFO / LRU RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA_DISPATCH=8001 PUERTO_ESCUCHA_INTERRUPT=8005
<i>Consola.config (A)</i>	<i>Consola.config (B)</i>
IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[160, 144, 128] TIEMPO_PANTALLA=750	IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[64, 32, 96, 80] TIEMPO_PANTALLA=750
<i>Memoria.config</i>	
PUERTO_ESCUCHA=8002 TAM_MEMORIA=4096 TAM_PAGINA=64	

```
ENTRADAS_POR_TABLA=4
RETARDO_MEMORIA=2500
ALGORITMO_REEMPLAZO=CLOCK
MARCOS_POR_PROCESO=8
RETARDO_SWAP=1000
PATH_SWAP=/home/utnso/swap
TAMANIO_SWAP=10240
```

Prueba Estabilidad General

Actividades

1. Iniciar los módulos.
2. Ejecutar los siguientes scripts.
 - a. ESTABILIDAD_1
 - b. ESTABILIDAD_2
 - c. ESTABILIDAD_3
 - d. ESTABILIDAD_4
 - e. ESTABILIDAD_5
 - f. ESTABILIDAD_6
3. Esperar que finalicen todas las consolas.

Resultados Esperados

- No se detectan esperas activas ni memory leaks.
- El proceso C finaliza por *Segment Fault*.

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_CPU=127.0.0.1 PUERTO_CPU_DISPATCH=8001 PUERTO_CPU_INTERRUPT=8005 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO GRADO_MAX_MULTIPROGRAMACION=4 DISPOSITIVOS_IO=[DISCO, IMPRESORA] TIEMPOS_IO=[50, 100] QUANTUM_RR=80	ENTRADAS_TLB=4 REEMPLAZO_TLB=FIFO / LRU RETARDO_INSTRUCCION=25 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA_DISPATCH=8001 PUERTO_ESCUCHA_INTERRUPT=8005
<i>Memoria.config</i>	
PUERTO_ESCUCHA=8002 TAM_MEMORIA=4096 TAM_PAGINA=64 ENTRADAS_POR_TABLA=16 RETARDO_MEMORIA=250 ALGORITMO_REEMPLAZO=CLOCK MARCOS_POR_PROCESO=8 RETARDO_SWAP=100 PATH_SWAP=/home/utnso/swap	

TAMANIO_SWAP=10240	
<i>Consola.config (A)</i>	<i>Consola.config (B)</i>
IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[1024] TIEMPO_PANTALLA=500	IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[1024] TIEMPO_PANTALLA=500
<i>Consola.config (C)</i>	<i>Consola.config (D)</i>
IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[512] TIEMPO_PANTALLA=500	IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[512] TIEMPO_PANTALLA=500
<i>Consola.config (E)</i>	<i>Consola.config (F)</i>
IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[768] TIEMPO_PANTALLA=500	IP_KERNEL=127.0.0.1 PUERTO_KERNEL=8000 SEGMENTOS=[768] TIEMPO_PANTALLA=500

Planilla de Evaluación - TP2C2022

Nombre del Grupo	Nota (Grupal)

Legajo	Apellido y Nombres	Nota (Coloquio)

Evaluador/es Práctica	
Evaluador/es Coloquio	

Observaciones:

Sistema Completo	
El deploy se hace compilando los módulos en las máquinas del laboratorio.	
Los procesos se ejecutan de forma simultánea y la cantidad de hilos y subprocesos en el sistema es la adecuada.	
Los procesos establecen conexiones TCP/IP.	
El sistema no registra casos de Espera Activa ni Memory Leaks.	
El log respeta los lineamientos de logs mínimos y obligatorios de cada módulo	
El sistema no requiere permisos de superuser (sudo/root) para ejecutar correctamente.	
El sistema no requiere de Valgrind o algún proceso similar para ejecutar correctamente.	
El sistema utiliza una sincronización determinista (no utiliza más sleeps de los solicitados en el TP).	

Módulo Kernel	
Recibe correctamente los sets de instrucciones de los procesos Consola.	
Respetar el grado de multiprogramación definido por archivo de configuración.	
Se respeta el diagrama de 5 estados y sus transiciones.	
El planificador de corto plazo respeta el orden de llegada de los procesos en FIFO.	
El planificador de corto plazo envía las interrupciones a la CPU ante los eventos definidos.	
Se respeta la planificación de los Dispositivos de I/O del Kernel, Teclado y Pantalla	
Se trata al evento de Page Fault como un bloqueo del proceso.	

Módulo CPU	
Respetar el ciclo de instrucción.	
Actualizar correctamente el PCB antes de devolverlo al kernel.	
Interpretar correctamente las instrucciones definidas.	
Respetar los tiempos de retardo para las operaciones.	
Realizar las traducciones de memoria siguiendo lo definido en el enunciado	
Las traducciones de memoria requieren de información de la memoria que se consulta al momento de realizar la traducción.	
Desalojar correctamente el PCB en ejecución al momento de recibir una interrupción.	
La TLB mantiene sus entradas al momento de cambiar el proceso de la CPU.	
La TLB respeta los algoritmos de reemplazo de sus entradas.	

Módulo Memoria

Respetar el tamaño definido para las páginas.

Respetar los límites de marcos por proceso.

Respetar los límites de entradas por tabla.

Respetar los algoritmos de reemplazo de páginas.

Responder correctamente a los mensajes de Kernel y CPU.

Implementar correctamente el archivo de SWAP.

Se cumplen los tiempos de retardo para los accesos a memoria y swap.