

Ingeniería en Sistemas de Información

T.U.K.I

Documento de pruebas

¡Mirá que te pruebo hermano!



Cátedra de Sistemas Operativos

Trabajo práctico Cuatrimestral

-1C2022 -
Versión 2.0

Índice

Índice	2
Versión de Cambios	3
Aclaraciones	4
Prueba Base	5
Actividades	5
Resultados Esperados	5
Configuración del sistema	5
Prueba Deadlock	6
Actividades	6
Resultados Esperados	6
Configuración del sistema	6
Prueba Errores	7
Actividades	7
Resultados Esperados	7
Configuración del sistema	7

Versión de Cambios

v1.0 (28/05/2023) Publicación Inicial Check Obligatorio

v1.1 (01/06/2023) Ajustada estimación inicial en la Prueba Base

v2.0 (08/07/2023) Publicación Pruebas Entrega Final

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/tuki-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

- I. Iniciar los módulos.
- II. Ejecutar 3 módulos consola con los siguientes scripts en el orden indicado:
 - A. BASE_1
 - B. BASE_2
 - C. BASE_2
- III. Esperar su finalización.
- IV. Cambiar el Algoritmo a HRRN y esperar la finalización.

Resultados Esperados

- Con el algoritmo FIFO los procesos finalizan en el orden A-B-C.
- Con el algoritmo HRRN finaliza el proceso A primero pero los procesos B y C deberían ejecutar 2 ráfagas cada uno antes de que vuelva a ejecutar A.

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_FILESYSTEM=127.0.0.1 PUERTO_FILESYSTEM=8003 IP_CPU=127.0.0.1 PUERTO_CPU=8001 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO ESTIMACION_INICIAL=2000 HRRN_ALFA=0.5 GRADO_MAX_MULTIPROGRAMACION=4 RECURSOS=[DISCO] INSTANCIAS_RECURSOS=[1]	RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8001 TAM_MAX_SEGMENTO=128
<i>Memoria.config</i>	<i>FileSystem.config</i>
PUERTO_ESCUCHA=8002 TAM_MEMORIA=4096 TAM_SEGMENTO_0=128 CANT_SEGMENTOS=16 RETARDO_MEMORIA=1000 RETARDO_COMPACTACION=60000 ALGORITMO_ASIGNACION=BEST	IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8003 RETARDO_ACCESO_BLOQUE=15000 PATH_SUPERBLOQUE=~ /fs/sbloque.dat PATH_BITMAP=~ /fs/bitmap.dat PATH_BLOQUES=~ /fs/bloques.dat PATH_FCB=~ /fs/fcb

Prueba Deadlock

Actividades

- I. Iniciar los módulos.
- II. Ejecutar 3 módulos consola con los siguientes scripts en el orden indicado:
 - A. DEADLOCK_1
 - B. DEADLOCK_2
 - C. DEADLOCK_3
- III. Esperar a que los tres procesos queden bloqueados.
- IV. Ejecutar una nueva consola con el script DEADLOCK_4.

Resultados Esperados

- Los procesos deberían tomar un recurso cada uno y mientras se encuentren en I/O otro proceso debería tomar el otro recurso que requieren dejando el sistema en un deadlock.
- El script DEADLOCK_4 nunca debería llegar a estado READY

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_FILESYSTEM=127.0.0.1 PUERTO_FILESYSTEM=8003 IP_CPU=127.0.0.1 PUERTO_CPU=8001 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO ESTIMACION_INICIAL=10000 HRRN_ALFA=0.5 GRADO_MAX_MULTIPROGRAMACION=3 RECURSOS=[DISCO, IMPRESORA, SCANNER] INSTANCIAS_RECURSOS=[1,1,1]	RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8001 TAM_MAX_SEGMENTO=128
<i>Memoria.config</i>	<i>FileSystem.config</i>
PUERTO_ESCUCHA=8002 TAM_MEMORIA=4096 TAM_SEGMENTO_0=128 CANT_SEGMENTOS=16 RETARDO_MEMORIA=1000 RETARDO_COMPACTACION=60000 ALGORITMO_ASIGNACION=BEST	IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8003 RETARDO_ACCESO_BLOQUE=15000 PATH_SUPERBLOQUE=~ /fs/sbloque.dat PATH_BITMAP=~ /fs/bitmap.dat PATH_BLOQUES=~ /fs/bloques.dat PATH_FCB=~ /fs/fcb

Prueba Memoria

Actividades

- I. Iniciar los módulos.
- II. Ejecutar 3 módulos consola con los siguientes scripts en el orden indicado:
 - A. MEMORIA_1
 - B. MEMORIA_2
 - C. MEMORIA_3
- III. Esperar a que los tres procesos finalicen.

Resultados Esperados

- Los procesos son capaces de leer y escribir en el segmento compartido.
- La memoria es capaz de compactar.

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_FILESYSTEM=127.0.0.1 PUERTO_FILESYSTEM=8003 IP_CPU=127.0.0.1 PUERTO_CPU=8001 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO ESTIMACION_INICIAL=10000 HRRN_ALFA=0.5 GRADO_MAX_MULTIPROGRAMACION=12 RECURSOS=[RA, RB, RC] INSTANCIAS_RECURSOS=[1, 0, 0]	RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8001 TAM_MAX_SEGMENTO=256
<i>Memoria.config</i>	<i>FileSystem.config</i>
PUERTO_ESCUCHA=8002 TAM_MEMORIA=1024 TAM_SEGMENTO_0=128 CANT_SEGMENTOS=16 RETARDO_MEMORIA=1000 RETARDO_COMPACTACION=60000 ALGORITMO_ASIGNACION=FIRST / BEST / WORST	IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8003 RETARDO_ACCESO_BLOQUE=15000 PATH_SUPERBLOQUE=~ / fs / sbloque.dat PATH_BITMAP=~ / fs / bitmap.dat PATH_BLOQUES=~ / fs / bloques.dat PATH_FCB=~ / fs / fcb

Prueba File System

Actividades

- I. Iniciar los módulos.
- II. Ejecutar 2 módulos consola con los siguientes scripts en el orden indicado:
 - A. FS_1
 - B. FS_2
- III. Esperar a que los procesos finalicen.
- IV. Finalizar todos los módulos.
- V. Iniciar nuevamente el sistema y ejecutar 4 consolas con el script FS_3.
- VI. Esperar la finalización de todos los scripts.

Resultados Esperados

- La compactación no inicia hasta que no finalizan las lecturas / escrituras pendientes en el FS.
- El Filesystem es capaz de obtener los datos guardados en una ejecución anterior.
- Los procesos se bloquean al intentar abrir un archivo que se encuentra abierto por otro proceso.

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_FILESYSTEM=127.0.0.1 PUERTO_FILESYSTEM=8003 IP_CPU=127.0.0.1 PUERTO_CPU=8001 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO ESTIMACION_INICIAL=10000 HRRN_ALFA=0.5 GRADO_MAX_MULTIPROGRAMACION=12 RECURSOS=[RA, RB] INSTANCIAS_RECURSOS=[1, 0]	RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8001 TAM_MAX_SEGMENTO=256
<i>Memoria.config</i>	
PUERTO_ESCUCHA=8002 TAM_MEMORIA=512 TAM_SEGMENTO_0=128 CANT_SEGMENTOS=16 RETARDO_MEMORIA=1000 RETARDO_COMPACTACION=30000	

ALGORITMO_ASIGNACION=FIRST	
<i>FileSystem.config</i>	<i>sbloque.dat</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8003 RETARDO_ACCESO_BLOQUE=15000 PATH_SUPERBLOQUE=~ /fs/sbloque.dat PATH_BITMAP=~ /fs/bitmap.dat PATH_BLOQUES=~ /fs/bloques.dat PATH_FCB=~ /fs/fcb	BLOCK_SIZE=64 BLOCK_COUNT=1024

Prueba Errores

Actividades

- I. Iniciar los módulos
- II. Ejecutar módulos consola con los siguientes scripts:
 - A. ERROR_1
 - B. ERROR_2
 - C. ERROR_3
 - D. ERROR_4

Resultados Esperados

- Todos los procesos deberían salir a Exit por errores. Verificar los logs.
 - ERROR_1: Wait de recurso no existente.
 - ERROR_2: Signal de recurso no existente.
 - ERROR_3: Segment Fault
 - ERROR_4: Out of Memory

Configuración del sistema

<i>Kernel.config</i>	<i>CPU.config</i>
IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 IP_FILESYSTEM=127.0.0.1 PUERTO_FILESYSTEM=8003 IP_CPU=127.0.0.1 PUERTO_CPU=8001 PUERTO_ESCUCHA=8000 ALGORITMO_PLANIFICACION=FIFO ESTIMACION_INICIAL=10000 HRRN_ALFA=0.5 GRADO_MAX_MULTIPROGRAMACION=4 RECURSOS=[DISCO] INSTANCIAS_RECURSOS=[1]	RETARDO_INSTRUCCION=1000 IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8001 TAM_MAX_SEGMENTO=256
<i>Memoria.config</i>	<i>FileSystem.config</i>
PUERTO_ESCUCHA=8002 TAM_MEMORIA=1024 TAM_SEGMENTO_0=128 CANT_SEGMENTOS=16 RETARDO_MEMORIA=1000 RETARDO_COMPACTACION=60000 ALGORITMO_ASIGNACION=BEST	IP_MEMORIA=127.0.0.1 PUERTO_MEMORIA=8002 PUERTO_ESCUCHA=8003 RETARDO_ACCESO_BLOQUE=15000 PATH_SUPERBLOQUE=~ /fs/sbloque.dat PATH_BITMAP=~ /fs/bitmap.dat PATH_BLOQUES=~ /fs/bloques.dat

	PATH_FCB=~/fs/fcb
--	-------------------

Planilla de Evaluación - TP1C2023

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 estima correctamente utilizando el Response Ratio de los procesos en Ready.	
El planificador de corto plazo envía las interrupciones a la CPU ante los eventos definidos.	
Se respeta la cantidad de recursos definidos por archivo de configuración	
Se respetan las operaciones de FS al momento de solicitar una compactación a la Memoria	
Se trata correctamente a los archivos como un recursos de 1 sola instancia.	
La compactación espera adecuadamente a la hora de encontrar operaciones de F_READ / F_WRITE en curso.	
Se respetan los tiempos de I/O que se reciben de la CPU.	

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 se realizan correctamente.	
Se validan correctamente los tamaños de segmento y se informan de los errores	

Módulo Memoria	
Respetar el tamaño definido para el segmento 0	
Respetar el límite de cantidad máxima de segmentos	
Al compactar mover los datos hacia su nueva posición y no se pierde información.	
Respetar el criterio de First Fit a la hora de elegir un hueco libre.	
Respetar el criterio de Best Fit a la hora de elegir un hueco libre.	
Respetar el criterio de Worst Fit a la hora de elegir un hueco libre.	
Se cumplen los tiempos de retardo para los accesos a memoria y compactación.	

Módulo File System	
Respetar el tamaño de bloques definido en el Superbloque	
Actualizar correctamente el bitmap de bloques a la hora de asignar bloques a los archivos.	
Mantiene actualizados los FCB de los archivos.	
Permite iniciar utilizando un estado anterior y mantiene los datos grabados.	
Permite leer y escribir archivos interactuando con la memoria.	