

Ingeniería en Sistemas de Información

A-MongOs

Documento de pruebas

Yo estaba en electricidad y después fui al reactor!!



Cátedra de Sistemas Operativos

Trabajo práctico Cuatrimestral

-1C2021 -
Versión 1.2

Índice

Índice	2
Versión de Cambios	4
Aclaraciones	5
Estabilidad General	6
Objetivo	6
Actividades	6
Resultados Esperados	6
Configuración del sistema	6
Discordador - CPU	7
Actividades	7
Resultados Esperados	7
Configuración del sistema	7
Discordador - E/S	8
Actividades	8
Resultados Esperados	8
Configuración del sistema	8
Memoria - Segmentación	9
Actividades	9
Resultados Esperados	9
Configuración del sistema	10
Memoria - Paginación	11
Actividades	11
Resultados Esperados	11
Configuración del sistema	11
File System	13
Actividades	13
Resultados Esperados	13
Configuración del sistema	13

File System - Sabotajes	14
Actividades	14
Resultados Esperados	14
Configuración del sistema	14

Versión de Cambios

v1.0 (12/07/2021) Publicación Inicial

v1.1 (16/07/2021) Se ajusta quantum de RR en prueba de paginación de 1 a 5

v1.2 (22/07/2021) Se ajusta la descripción de la prueba de Paginación para ser más claros.

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 detallados de las tareas que se utilizan en este documento se encuentran subidos al repositorio correspondiente: <https://github.com/sisoputnfrba/a-mongos-pruebas> dentro de la carpeta Finales y al momento de nombrarlos en las pruebas solamente vamos a nombrarlos por su nombre y es responsabilidad de cada grupo completar con el path en caso de ser necesario para encontrar el archivo correspondiente.

Por un tema de simplicidad en el enunciado de las pruebas, a las patotas se las va a nombrar por el nombre del archivo de tareas que deban realizar.

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).

La compilación debe hacerse en la máquina virtual de la cátedra en su edición Server y es un buen momento del cuatrimestre para que empiecen a familiarizarse con la VM en su edición Server y que ante cualquier duda o inconveniente se acerquen a los sábados de soporte o creen issues en el foro.

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

Se utilizará el release v1.0.0 del repositorio de [scripts de pruebas](#).

Los ayudantes podrán modificar las pruebas en caso de ser necesario.

Está totalmente prohibido subir archivos binarios al repositorio.

Estabilidad General

Objetivo

- Verificar la estabilidad general del sistema
- Controlar que no se tengan esperas activas ni memory leaks

Actividades

- Iniciar los módulos en el orden definido por el grupo en la primer vuelta con el algoritmo de planificación FIFO y en la segunda con RR
- Ejecutar los siguientes comandos en la consola del Discordiador.

```
> INICIAR_PATOTA 3 ES3_Patota1.txt 9|9 0|0 5|5
> INICIAR_PATOTA 3 ES3_Patota2.txt 4|0 2|6 8|2
> INICIAR_PATOTA 3 ES3_Patota3.txt 2|3 5|8 5|3
> INICIAR_PATOTA 3 ES3_Patota4.txt 0|9 4|4 9|0
> INICIAR_PATOTA 3 ES3_Patota5.txt 0|2 9|6 3|5
> INICIAR_PLANIFICACION
> LISTAR_TRIPULANTES
```

- Esperar el final de la planificación listando los tripulantes cada cierto tiempo.

Resultados Esperados

1. Deberán terminar todos los tripulantes.
2. No deben encontrarse ni esperas activas ni memory leaks.
3. Las patotas 3 y 4 deberán ejecutar sin verse afectadas por los encolamientos en E/S.

Configuración del sistema

Discordiador.config	Mi-RAM-HQ.config
GRADO_MULTITAREA=2 ALGORITMO=RR QUANTUM=1 DURACION_SABOTAJE=10 RETARDO_CICLO_CPU=1	TAMANIO_MEMORIA=16384 ESQUEMA_MEMORIA=SEGMENTACION TAMANIO_PAGINA=64 TAMANIO_SWAP=16384 ALGORITMO_REEMPLAZO=LRU CRITERIO_SELECCION=FF
i-Mongo-Store.config	Datos para el superbloque
TIEMPO_SINCRONIZACION=12 POSICIONES_SABOTAJE=[1 1]	BLOCK_SIZE=256 BLOCKS=32768

Discordiator - CPU

Actividades

- Iniciar los módulos en el orden definido por el grupo en la primer vuelta con el algoritmo de planificación FIFO y en la segunda con RR en Discordiator
- Ejecutar los siguientes comandos en la consola del Discordiator.

```
> INICIAR_PATOTA 2 CPU_Patota1.txt 1|1
> INICIAR_PATOTA 1 CPU_Patota2.txt 1|0
> INICIAR_PLANIFICACION
> LISTAR_TRIPULANTES
```

- Esperar que transcurran 5 ciclos de CPU.

```
> DETENER_PLANIFICACION
> INICIAR_PATOTA 1 CPU_Patota3.txt 0|1
> INICIAR_PLANIFICACION
> LISTAR_TRIPULANTES
```

- Esperar el fin de todas las tareas de los tripulantes.

Resultados Esperados

1. Deberán terminar todos los tripulantes
2. En el caso de FIFO una vez que un tripulante empieza a ejecutar nunca debe dejar de estar en ejecución hasta que termine sus tareas.
3. En el caso de RR los tripulantes deberán ir alternándose cada 3 ciclos de CPU.

Configuración del sistema

Discordiator.config	Mi-RAM-HQ.config
GRADO_MULTITAREA=2 ALGORITMO=FIFO / RR QUANTUM=3 DURACION_SABOTAJE=10 RETARDO_CICLO_CPU=1	TAMANIO_MEMORIA=8192 ESQUEMA_MEMORIA=SEGMENTACION TAMANIO_PAGINA=64 TAMANIO_SWAP=16384 ALGORITMO_REEMPLAZO=LRU CRITERIO_SELECCION=FF
i-Mongo-Store.config	Datos para el superbloque
TIEMPO_SINCRONIZACION=12 POSICIONES_SABOTAJE=[1 1]	BLOCK_SIZE=64 BLOCKS=16384

Discordiator - E/S

Actividades

- Iniciar los módulos en el orden definido por el grupo en la primer vuelta con el algoritmo de planificación FIFO y en la segunda con RR en Discordiator
- Ejecutar los siguientes comandos en la consola del Discordiator.

```
> INICIAR_PATOTA 2 IO_Patota1.txt 1|1
> INICIAR_PATOTA 1 IO_Patota2.txt 1|0
> INICIAR_PATOTA 1 IO_Patota3.txt 0|1
> INICIAR_PLANIFICACION
> LISTAR_TRIPULANTES
```

- Esperar el fin de todas las tareas de los tripulantes.

Resultados Esperados

1. Todos los tripulantes terminan su ejecución.
2. En FIFO al llegar los tripulantes de la patota 1 a su tarea de E/S se van a encolar y van a generar que los tripulantes de las otras 2 patotas tengan que esperar.
3. En RR debe llegar primero el tripulante de la patota 2 a realizar su tarea de E/S antes que los tripulantes de la patota 1.
4. Se liberan los bloques del archivo Basura.ims generado por el tripulante de la patota 2.

Configuración del sistema

Discordiator.config	Mi-RAM-HQ.config
GRADO_MULTITAREA=2 ALGORITMO=FIFO / RR QUANTUM=3 DURACION_SABOTAJE=10 RETARDO_CICLO_CPU=1	TAMANIO_MEMORIA=8192 ESQUEMA_MEMORIA=SEGMENTACION TAMANIO_PAGINA=64 TAMANIO_SWAP=16384 ALGORITMO_REEMPLAZO=LRU CRITERIO_SELECCION=FF
i-Mongo-Store.config	Datos para el superbloque
TIEMPO_SINCRONIZACION=12 POSICIONES_SABOTAJE=[1 1]	BLOCK_SIZE=64 BLOCKS=16384

Memoria - Segmentación

Actividades

- Iniciar los módulos en el orden definido por el grupo, en la primer vuelta configurando el algoritmo en First Fit (FF) y en la segunda corrida con Best Fit (BF) en Mi-RAM HQ
- En el módulo discordiador ejecutar los siguientes comandos

```
> INICIAR_PATOTA 4 SEG_PatotaA.txt  
> INICIAR_PATOTA 1 SEG_PatotaC.txt  
> INICIAR_PLANIFICACION  
> LISTAR_TRIPULANTES
```

- Enviar la señal para el DUMP de la memoria.
- Una vez iniciada la planificación esperar 10 ciclos de CPU y ejecutar los siguientes comandos.

```
> EXPULSAR_TRIPULANTE A1 (Tripulante 1 - Patota A)  
> EXPULSAR_TRIPULANTE A3 (Tripulante 3 - Patota A)  
> LISTAR_TRIPULANTES  
> INICIAR_PATOTA 2 SEG_PatotaB.txt  
> LISTAR_TRIPULANTES
```

- Enviar nuevamente la señal para el DUMP de la memoria.
- Esperar el fin de las tareas de los tripulantes.
- Enviar nuevamente la señal para el DUMP de la memoria.
- Enviar la señal para compactar la memoria
- Enviar nuevamente la señal para el DUMP de la memoria.

Resultados Esperados

1. Al momento de finalizar la patota C quedan varios segmentos libres correspondientes a las tareas, TCB y PCB.
2. Al momento de iniciar la patota B:
 - a. Con FF debe quedar un hueco libre al lado del PCB ya que debería ocupar el espacio del tripulante A1.
 - b. Con BF el PCB de la patota B ocupará el espacio del PCB de la patota C por lo que no habrá un hueco en ese lugar
3. Al momento de cargar las tareas se debe compactar la memoria para generar el hueco libre para las tareas de la patota B.
4. Los tripulantes finalizan su ejecución y quedan todos los segmentos libres.
5. Al compactar la memoria los segmentos se unifican y se pueden validar al ver el log del dump de la memoria.

Configuración del sistema

Discordiador.config	Mi-RAM-HQ.config
GRADO_MULTITAREA=2 ALGORITMO=RR QUANTUM=3 DURACION_SABOTAJE=10 RETARDO_CICLO_CPU=2	TAMANIO_MEMORIA=256 ESQUEMA_MEMORIA=SEGMENTACION TAMANIO_PAGINA=32 TAMANIO_SWAP=4096 ALGORITMO_REEMPLAZO=LRU CRITERIO_SELECCION=FF/BF
i-Mongo-Store.config	Datos para el superbloque
TIEMPO_SINCRONIZACION=10 POSICIONES_SABOTAJE=[1 1]	BLOCK_SIZE=64 BLOCKS=16384

Memoria - Paginación

Actividades

- Iniciar los módulos en el orden definido por el grupo, en la primer vuelta ejecutando el algoritmo de LRU y en la segunda con el algoritmo CLOCK en Mi-RAM HQ
- En el módulo discordiador crear las siguientes patotas e inicia la planificación

```
> INICIAR_PATOTA 1 PAG_PatotaA.txt 1|1  
> INICIAR_PATOTA 1 PAG_PatotaB.txt 3|3  
> INICIAR_PATOTA 1 PAG_PatotaC.txt 5|5  
> INICIAR_PLANIFICACION
```

- Controlar los reemplazos de páginas al momento de guardar los datos mientras se inicia la patota y sus tripulantes.
- Mientras ejecutan los tripulantes verificar cómo se realizan las peticiones de memoria y controlar los reemplazos de página para obtener las tareas.
- Finalizar todos los tripulantes y verificar que la memoria queda completamente limpia.

Resultados Esperados

1. Al llegar las 3 patotas a Ready, Patota A y Patota B deberían estar en swap, y patota C ocupar la totalidad de los marcos
2. Ejecuta el tripulante de la patota A, reemplazando dos páginas de C
3. Ejecuta el tripulante de la patota B, reemplazando dos páginas de C
4. Ejecuta el tripulante de la patota C, reemplazando dos páginas de A
5. Ejecuta el tripulante de la patota A, reemplazando dos páginas de B
6. Ejecuta el tripulante de la patota B, reemplazando dos páginas de C
7. Ejecuta el tripulante de la patota C, reemplazando dos páginas de A
8. Ejecuta el tripulante de la patota A, reemplazando dos páginas de B
9. Finaliza A
10. Ejecuta el tripulante de la patota B, aprovechando los frames dejados por A
11. Finaliza B
12. C utiliza todas las páginas de memoria hasta finalizar.

Configuración del sistema

Discordiador.config	Mi-RAM-HQ.config
GRADO_MULTITAREA=1 ALGORITMO= RR QUANTUM=5 DURACION_SABOTAJE=10 RETARDO_CICLO_CPU=2	TAMANIO_MEMORIA=128 ESQUEMA_MEMORIA=PAGINACION TAMANIO_PAGINA=32 TAMANIO_SWAP=16384 ALGORITMO_REEMPLAZO=LRU/CLOCK CRITERIO_SELECCION=FF

i-Mongo-Store.config	Datos para el superbloque
TIEMPO_SINCRONIZACION=10 POSICIONES_SABOTAJE=[1 1]	BLOCK_SIZE=64 BLOCKS=16384

File System

Actividades

- Iniciar los módulos en el orden definido por el grupo
- En el módulo discordiador crear las siguientes patotas e inicia la planificación

```
> INICIAR_PATOTA 3 FS_PatotaA.txt  
> INICIAR_PATOTA 3 FS_PatotaB.txt  
> INICIAR_PLANIFICACION
```

- Controlar las escrituras de los recursos en los bloques
- Mientras ejecutan los tripulantes verificar cómo se realizan las peticiones de I/O y que se bloqueen los tripulantes en el orden y tiempo estipulado
- Finalizar todos los tripulantes y verificar que la sincronización de datos sea correcta
- Volver a levantar el filesystem y correr esta prueba, y verificar que la información no se sobreescriba

Resultados Esperados

1. El File system se formatea y crea correctamente
2. Se crean correctamente 60 instancias de cada recurso al terminar la primera ejecución
3. Se crean correctamente 6 bitácoras
4. Al tirar abajo el módulo y volver a levantarlo, esos recursos deben seguir ahí
5. Luego de la segunda ejecución debería haber 120 instancias de cada recurso

Configuración del sistema

Discordiador.config	Mi-RAM-HQ.config
GRADO_MULTITAREA=3 ALGORITMO= FIFO QUANTUM=1 DURACION_SABOTAJE=10 RETARDO_CICLO_CPU=1	TAMANIO_MEMORIA=2048 ESQUEMA_MEMORIA=SEGMENTACION TAMANIO_PAGINA=32 TAMANIO_SWAP=16384 ALGORITMO_REEMPLAZO=LRU CRITERIO_SELECCION=FF
i-Mongo-Store.config	Datos para el superbloque
TIEMPO_SINCRONIZACION=5 POSICIONES_SABOTAJE=[1 1]	BLOCK_SIZE=32 BLOCKS=16384

File System - Sabotajes

Actividades

- Iniciar los módulos en el orden definido por el grupo utilizando el mismo FS de la prueba de File System después de haber ejecutado las pruebas.
- En el módulo discordiador crear las siguientes patotas e inicia la planificación

```
> INICIAR_PATOTA 1 FSCK_PatotaA.txt 0|0
> INICIAR_PATOTA 1 FSCK_PatotaB.txt 8|0
> INICIAR_PATOTA 1 FSCK_PatotaC.txt 8|8
> INICIAR_PATOTA 1 FSCK_PatotaD.txt 0|8
> INICIAR_PLANIFICACION
```

- Realizar un sabotaje (definido por el ayudante)
- Enviar la señal de sabotaje y esperar la resolución del sabotaje
- El ayudante podrá repetir esta prueba cuantas veces considere necesarias sin necesidad de reiniciar el módulo.

Resultados Esperados

1. Se detectan y resuelven correctamente los sabotajes.

Configuración del sistema

Discordiador.config	Mi-RAM-HQ.config
GRADO_MULTITAREA=1 ALGORITMO=FIFO QUANTUM=1 DURACION_SABOTAJE=10 RETARDO_CICLO_CPU=1	TAMANIO_MEMORIA=2048 ESQUEMA_MEMORIA=SEGMENTACION TAMANIO_PAGINA=32 TAMANIO_SWAP=16384 ALGORITMO_REEMPLAZO=LRU CRITERIO_SELECCION=FF
i-Mongo-Store.config	Datos para el superbloque
TIEMPO_SINCRONIZACION=1 POSICIONES_SABOTAJE=[1 1,7 1,7 7,1 7,4 4]	BLOCK_SIZE=32 BLOCKS=16384

Planilla de Evaluación - TP1C2021

Sistema Completo	
El deploy se hace en un tiempo límite de 15 a 20 minutos.	
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 y se comunican mediante un protocolo.	
El sistema no registra casos de Espera Activa ni Memory Leaks.¹	
El log permite determinar en todo momento el estado actual y anterior de los diversos procesos y del sistema junto con sus cambios significativos.	
El sistema continúa su funcionamiento ante comandos erróneos o paths inexistentes (informando al usuario el error).	
<i>El sistema no requiere permisos de superuser (sudo/root) para ejecutar correctamente.</i>	

Proceso Discordador	
Se respeta la planificación bajo el algoritmo FIFO	
Se respeta la planificación bajo el algoritmo RR viéndose la alternancia en el estado EXEC de los tripulantes.	
Las peticiones de Entrada / Salida se encolan ya que hay 1 solo dispositivo de E/S	
Se respetan los tiempos de retardo de CPU.	
Se respetan los tiempos de espera en cola de bloqueados.	
Se respeta el grado de multitarea.	
Se respeta el tiempo de sabotaje.	

Proceso Mi-RAM HQ	
Se respeta el manejo de peticiones de manera concurrente.	
Se mantiene actualizado el Mapa.*	
Se respeta el algoritmo de selección de huecos First Fit (FF).	
Se respeta el algoritmo de selección de huecos Best Fit (BF).	
Se respeta el algoritmo de reemplazo de páginas LRU.	
Se respeta el algoritmo de reemplazo de páginas CLOCK.	
Se respeta la utilización del archivo de swap.	

¹ Una espera activa o un memory leak serán motivo de desaprobación directa.

Proceso i-Mongo Store	
Se respeta el manejo de peticiones de manera concurrente	
Permite crear, escribir, leer y borrar archivos	
Las estructuras del FileSystem se actualizan correctamente ante cambios en los archivos y directorios (Bitmap, metadata del archivo, etc)	
Ante cambios en el contenido del archivo se actualiza correctamente el MD5	
Utiliza correctamente la especificación de FileSystem indicada	
Se respeta el tiempo de sincronización.	
Los sabotajes se detectan y resuelven correctamente	