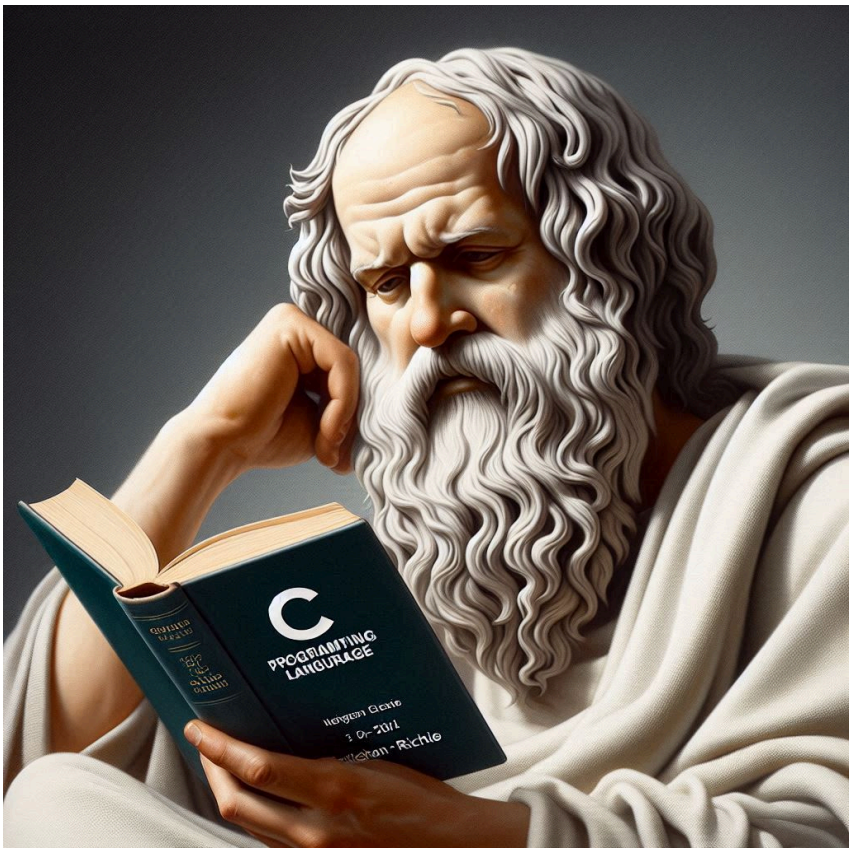
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Ingeniería en Sistemas de Información

C-Comenta

Documento de pruebas

Solo C que no C nada



Cátedra de Sistemas Operativos

Trabajo práctico Cuatrimestral

-1C2024 -
Versión 1.2

Índice

Versión de Cambios

v1.0 (03/07/2024) Publicación Inicial de Pruebas Finales
v1.1 (04/07/2024) Ajustes de redacción.
v1.2 (10/07/2024) Aumentado el tamaño de memoria en Salvation's Edge

Criterios de Evaluación

Los grupos deberán concurrir al laboratorio habiendo corrido las pruebas y siendo conscientes de que las mismas funcionan en un entorno distribuido, es decir, **si el trabajo práctico no puede correr en más de una máquina el mismo no se evaluará**.

Al momento de realizar la evaluación en el laboratorio los alumnos dispondrán de un máximo de **10 minutos** para configurar el ambiente y validar que las conexiones se encuentren funcionando, caso contrario se considerará que el grupo no se encuentra en condiciones de ser evaluador.

Los grupos contarán con **una única instancia de evaluación por fecha**, es decir, que ante un error no resoluble en el momento, se considerará que el grupo no puede continuar la evaluación y por lo tanto esa entrega se encuentra **desaprobada**.

Aclaraciones

Todos los scripts para realizar las pruebas que se enumeran en este documento se encuentran subidos al repositorio: c-comenta-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 computadora o del desarrollo de 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.

En las configuraciones de las Interfaces el nombre de la interfaz es el nombre del archivo .config de la misma, es decir, si el config se llama ***IO_1.config*** el nombre de la interfaz deberá ser ***IO_1***

Está totalmente prohibido subir archivos binarios al repositorio.

Prueba Planificación

Actividades

1. Iniciar los módulos.
2. Ejecutar los siguientes comandos en la consola del Kernel
    a. `EJECUTAR_SCRIPT /scripts_kernel/PRUEBA_PLANI`
3. Esperar a que empiece a ejecutar PLANI_4 y ejecutar `FINALIZAR_PROCESO` con el PID de PLANI_4.
4. Esperar que finalicen los demás procesos y cortar la prueba.
5. Cambiar el algoritmo de planificación a RR y volver a ejecutar en la consola del Kernel
    a. `EJECUTAR_SCRIPT /scripts_kernel/PRUEBA_PLANI`
6. Esperar a que finalicen 3 procesos y ejecutar `FINALIZAR_PROCESO` con el PID de PLANI_4.
7. Finalizar la prueba, cambiar el algoritmo de planificación a VRR y volver a ejecutar.
    a. `EJECUTAR_SCRIPT /scripts_kernel/PRUEBA_PLANI`
8. Esperar a que finalicen 3 procesos y finalizar la prueba.

Resultados Esperados

- 3 de los 4 procesos finalizan sin problemas
- En FIFO Para que puedan volver a ejecutar PLANI_2 y PLANI_3 hay que matar el proceso PLANI_4
- En RR finalizan PLANI_1, luego PLANI_3 (el cual es desalojado 2 veces por fin de quantum) y por último PLANI_2. PLANI_4 continúa ejecutando.
- En VRR finalizan en el mismo orden que RR, pero PLANI_3 es desalojado 3 veces por fin de quantum.

Configuración del sistema

Kernel.config	*CPU.config*
ALGORITMO_PLANIFICACION=FIFO QUANTUM=2750 RECURSOS=[RECURSO] INSTANCIAS_RECURSOS=[1] GRADO_MULTIPROGRAMACION=10	CANTIDAD_ENTRADAS_TLB=32 ALGORITMO_TLB=FIFO
Memoria.config	*SLP1.config*
TAM_MEMORIA=1024 TAM_PAGINA=32 RETARDO_RESPUESTA=1000	TIPO_INTERFAZ=GENERICA TIEMPO_UNIDAD_TRABAJO=50

Prueba Deadlock

Actividades

1. Iniciar los módulos.
2. Ejecutar los siguientes comandos en la consola del Kernel
    a. `EJECUTAR_SCRIPT /scripts_kernel/PRUEBA_DEADLOCK`
3. Esperar a que los 4 procesos se queden bloqueados.
4. Finalizar un proceso por consola del Kernel.

Resultados Esperados

- Los 4 procesos se quedan en deadlock
- Al finalizar un proceso los demás se liberan

Configuración del sistema

Kernel.config	*CPU.config*
ALGORITMO_PLANIFICACION=FIFO QUANTUM=1500 RECURSOS=[RA, RB, RC, RD] INSTANCIAS_RECURSOS=[1, 1, 1, 1] GRADO_MULTIPROGRAMACION=10	CANTIDAD_ENTRADAS_TLB=0 ALGORITMO_TLB=FIFO
Memoria.config	*ESPERA.config*
TAM_MEMORIA=1024 TAM_PAGINA=32 RETARDO_RESPUESTA=1000	TIPO_INTERFAZ=GENERICA TIEMPO_UNIDAD_TRABAJO=500

Prueba Memoria y TLB

Actividades

1. Iniciar los módulos.
2. Ejecutar los siguientes comandos en la consola del Kernel
    a. `INICIAR_PROCESO /scripts_memoria/MEMORIA_1`
3. Esperar a que finalice el proceso.
4. Finalizar todos los módulos, cambiar el algoritmo de reemplazo de TLB por LRU y volver a iniciar todo.
5. Ejecutar los siguientes comandos en la consola del Kernel
    a. `INICIAR_PROCESO /scripts_memoria/MEMORIA_1`
6. Esperar a que finalice el proceso.
7. Ejecutar los siguientes comandos en la consola del Kernel
    a. `INICIAR_PROCESO /scripts_memoria/MEMORIA_2`
8. Esperar a que finalice el proceso.
9. Ejecutar los siguientes comandos en la consola del Kernel
    a. `INICIAR_PROCESO /scripts_memoria/MEMORIA_3`
10. Esperar a que finalice el proceso.

Resultados Esperados

- En las ejecuciones del proceso MEMORIA_1 se observan diferencias en los reemplazos de la TLB.
- En la ejecución del proceso MEMORIA_2 se observa que puede recuperar correctamente el valor de la memoria que se encuentra en 2 páginas diferentes.
- En la ejecución del proceso MEMORIA_3 el mismo debería finalizar por error de *Out of Memory*.

Configuración del sistema

Kernel.config	*CPU.config*
ALGORITMO_PLANIFICACION=FIFO QUANTUM=2000 RECURSOS=[RECURSO] INSTANCIAS_RECURSOS=[1] GRADO_MULTIPROGRAMACION=10	CANTIDAD_ENTRADAS_TLB=4 ALGORITMO_TLB=FIFO
Memoria.config	*IO_GEN_SLEEP.config*
TAM_MEMORIA=1024 TAM_PAGINA=32 RETARDO_RESPUESTA=1000	TIPO_INTERFAZ=GENERICA TIEMPO_UNIDAD_TRABAJO=250

Prueba IO

Actividades

1. Iniciar los módulos.
2. Ejecutar los siguientes comandos en la consola del Kernel
    a. `EJECUTAR_SCRIPT /scripts_kernel/PRUEBA_IO`
3. Ingresar los siguientes textos sin comillas:
    a. Para IO_A, "`WAR NEVER CHANGES...`"
    b. Para IO_C, "`Sistemas Operativos 1c2024`"
4. Esperar a que finalicen 3 procesos y cortar la prueba.

Resultados Esperados

- El proceso IO_A debería imprimir la frase: "WAR, WAR NEVER CHANGES…"
- El proceso IO_B debería imprimir la frase: "I don't want to set the world on fire"
- El proceso IO_C debería imprimir la frase: "Sistemas Operativos 1c2024"

Configuración del sistema

Kernel.config	*Memoria.config*	*CPU.config*
ALGORITMO_PLANIFICACION=RR QUANTUM=750 RECURSOS=[REC1] INSTANCIAS_RECURSOS=[1] GRADO_MULTIPROGRAMACION=10	TAM_MEMORIA=1024 TAM_PAGINA=16 RETARDO_RESPUESTA=100	CANTIDAD_ENTRADAS_TLB=0 ALGORITMO_TLB=FIFO

GENERICA.config	*TECLADO.config*	*MONITOR.config*
TIPO_INTERFAZ=GENERICA TIEMPO_UNIDAD_TRABAJO=250	TIPO_INTERFAZ=STDIN TIEMPO_UNIDAD_TRABAJO=250	TIPO_INTERFAZ=STDOUT TIEMPO_UNIDAD_TRABAJO=250

Prueba FS

Actividades

1. Iniciar los módulos.
2. Ejecutar los siguientes comandos en la consola del Kernel
    a. `INICIAR_PROCESO /scripts_memoria/FS_1`
    b. `INICIAR_PROCESO /scripts_memoria/FS_2`
3. Ingresar para FS_1 el siguiente texto sin comillas “`Fallout 1 Fallout 2 Fallout 3 Fallout: New Vegas Fallout 4 Fallout 76`”
4. Esperar a que finalicen los 2 procesos y cortar la prueba finalizando la ejecución de todos los módulos.
5. Iniciar nuevamente los módulos.
6. Ejecutar los siguientes comandos en la consola del Kernel
    a. `INICIAR_PROCESO /scripts_memoria/FS_3`
    b. `INICIAR_PROCESO /scripts_memoria/FS_4`
7. Esperar a que finalice el proceso y cortar la prueba.

Resultados Esperados

- El proceso FS_1 genera 2 archivos y escribe en ellos.
- El proceso FS_2 genera 2 archivos de tamaño 10 bytes.
- El proceso FS_3 lee el contenido del primer archivo creado por FS_1.
- El proceso FS_4 crea un archivo de 250 bytes, elimina el primer archivo creado por FS_2 y al querer ampliar el segundo archivo de los creados por FS_2 requiere de compactar.

Configuración del sistema

Kernel.config	*Memoria.config*	*CPU.config*
ALGORITMO_PLANIFICACION=RR QUANTUM=5000 RECURSOS=[REC1] INSTANCIAS_RECURSOS=[1] GRADO_MULTIPROGRAMACION=10	TAM_MEMORIA=1024 TAM_PAGINA=16 RETARDO_RESPUESTA=100	CANTIDAD_ENTRADAS_TLB=0 ALGORITMO_TLB=FIFO

FS.config	*TECLADO.config*	*MONITOR.config*
TIPO_INTERFAZ=DIALFS TIEMPO_UNIDAD_TRABAJO=2000 BLOCK_SIZE=16 BLOCK_COUNT=32 RETRASO_COMPACTACION=7500	TIPO_INTERFAZ=STDIN TIEMPO_UNIDAD_TRABAJO=250	TIPO_INTERFAZ=STDOUT TIEMPO_UNIDAD_TRABAJO=250

Prueba Salvation's Edge

Actividades

1. Iniciar los módulos.
2. Ejecutar los siguientes comandos en la consola del Kernel
    a. `EJECUTAR_SCRIPT /scripts_kernel/PRUEBA_SALVATIONS_EDGE`
3. Esperar que empiecen a ejecutar los 2 primeros scripts.
4. Ejecutar en la consola del Kernel el comando `MULTIPROGRAMACION 100`
5. Esperar a que finalicen todos los procesos salvo aquellos que están en un loop infinito.

Aclaraciones

1. Ir ejecutando los comandos que el ayudante evaluador indique.
2. A intervalos regulares ejecutar los siguientes comandos en la consola del Kernel
    a. `DETENER_PLANIFICACION`
    b. `PROCESO_ESTADO`
    c. `INICIAR_PLANIFICACION`

Resultados Esperados

- No se observan esperas activas y/o memory leaks.
-

Configuración del sistema

Kernel.config	*Memoria.config*	*CPU.config*
ALGORITMO_PLANIFICACION=VRR QUANTUM=500 RECURSOS=[RA, RB, RC, RD] INSTANCIAS_RECURSOS=[1,1,1,1] GRADO_MULTIPROGRAMACION=3	TAM_MEMORIA=4096 TAM_PAGINA=32 RETARDO_RESPUESTA=100	CANTIDAD_ENTRADAS_TLB=16 ALGORITMO_TLB=LRU

GENERICA.config	*SLP1.config*
TIPO_INTERFAZ=GENERICA TIEMPO_UNIDAD_TRABAJO=250	TIPO_INTERFAZ=GENERICA TIEMPO_UNIDAD_TRABAJO=50
ESPERA.config	*TECLADO.config*
TIPO_INTERFAZ=GENERICA TIEMPO_UNIDAD_TRABAJO=500	TIPO_INTERFAZ=STDIN TIEMPO_UNIDAD_TRABAJO=50
MONITOR.config	
TIPO_INTERFAZ=STDOUT TIEMPO_UNIDAD_TRABAJO=50	

Planilla de Evaluación - TP1C2024

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 en menos de 15 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.	
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 determinística (no utiliza más sleeps de los solicitados).	

Módulo Kernel	
Interpreta correctamente los comandos introducidos por su consola.	
Respeta 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 ejecuta correctamente las interrupciones por fin de quantum.	
El planificador de corto plazo respeta la cola de ready prioritaria en VRR	
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.	

Módulo CPU	
Respeta el ciclo de instrucción.	
Actualiza correctamente el PCB antes de devolverlo al kernel.	
Interpreta correctamente las instrucciones definidas.	
Realiza las traducciones de dirección lógica a físicasiguiendo lo definido en el enunciado.	
Los accesos a memoria se realizan correctamente.	

Módulo Memoria	
Se respetan los tamaños de página.	
Se respetan los retardos en las operaciones.	
Se administra correctamente el espacio de usuario utilizando un único **void***	
Permite la creación y finalización de procesos	
Permite acceder correctamente a las tablas de páginas.	
Permite acceder al espacio de usuario únicamente a través de direcciones físicas.	

Módulo Interfaz I/O	
Respeta correctamente la especificación de las instrucciones que acepta	
Respeta la especificación de los accesos a Memoria.	
DialFS: Permite la creación de archivos	
DialFS: Permite la eliminación de archivos	
DialFS: Permite el truncado de archivos	
DialFS: Permite la lectura y escritura de archivos	
DialFS: Compacta los archivos	