

# Ráfaga - Fe de errata

## 06/12/2014

- `WAIT` Toma el valor apuntado por el registro B y evalúa si este contador es negativo o cero. De serlo, invoca a la operación de `BLOK`; de lo contrario, resta una unidad y guarda en la memoria correspondiente el nuevo valor. Altera el registro A por uso interno.

## 16/11/2014

- La segunda aclaración del 03/11/2014 esta incompleta, debería leerse así: Cuando la CPU reciba un TCB en modo Kernel (KM activado) deberá realizar todas las **lecturas/escrituras** sobre la MSP utilizando el PID default del Kernel (ver aclaración más arriba), y no el PID indicado por el TCB en cuestión. **Las únicas excepciones son OUTC e INNC, que actúan sobre la memoria del proceso de usuario que invocó la llamada al sistema.**

## 03/11/2014

- El PID del TCB en modo Kernel (TCB KM) deberá ser conocido por los Procesos Kernel y CPU, por ejemplo PID = 0 (cero). De esta forma la CPU puede acceder al espacio de direcciones del TCB KM cuando lo requiera.
- ~~Quando la CPU reciba un TCB en modo Kernel (KM activado) deberá realizar todas las lecturas/escrituras sobre la MSP utilizando el PID default del Kernel (ver aclaración más arriba), y no el PID indicado por el TCB en cuestión.~~  
Editado el 16/11/2014, ver aclaración más arriba .
- Páginas 15, 16: En la sección de *Archivo de Configuración*, falta agregar un parámetro:

`RETARDO` (Retardo de Instrucción, numérico): Tiempo en milisegundos que el Proceso CPU debe esperar luego de ejecutar cada instrucción.

Por ejemplo: `RETARDO=1000` Indica un retardo de 1000 milisegundos.

## 18/09/2014

- Páginas 9, 12:

Se desbloqueará este TCB de usuario ... Cantidad de instrucciones que los hilos de usuario ...

Por TCBs de usuario e hilos de usuario no se hace referencia a ULTs (User Level Threads) si no a diferenciar el TCB del Kernel con los TCBs que ejecutan los usuarios.

- Pagina 8:

Ante la eventual desconexión de una CPU de un Proceso en ejecución, el planificador deberá notificar a la Consola del Proceso dicha excepción y abotar la ejecución

Cabe aclarar que, si fuera el TCB del Kernel el que estuviera ejecutando en esa CPU, el Kernel deberá finalizar su ejecución, y con él todos los Procesos.

- Páginas 5, 9, 13, 13, 29, 29:

"compilado", "compilar"

A lo largo de todo el enunciado, la palabra "compilado" deberá entenderse como "ensamblado", y "compilar" como "ensamblar".

- Pagina 18:

Solicitar Memoria [...] En caso que el pedido intente solicitar datos desde una posición de memoria inválida o que el mismo exceda los límites del segmento, retornará el correspondiente error de Violación de Segmento

(Segmentation Fault).

Un Segmentation Fault deberá abortar la ejecución de todos los TCBs del Proceso en cuestión.

- Pagina 25:

La explicación sobre los saltos de línea es errónea. Debería leerse:

JMPZ [Dirección] Altera el flujo de ejecución sólo si el valor del registro A es cero, para ejecutar la instrucción apuntada por la Dirección . El valor es el desplazamiento desde el inicio del programa.

JPNZ [Dirección] Altera el flujo de ejecución sólo si el valor del registro A **no es** cero, para ejecutar la instrucción apuntada por la Dirección . El valor es el desplazamiento desde el inicio del programa.

- Pagina 26:

En la operación CREA faltó aclarar:

De no tener espacio para crear el segmento de stack, deberá abortar el programa.

- Pagina 33:

La correcta fecha del Checkpoint número 5 es el Sabado 22 de Noviembre.

- Página 18:

Cuando la MSP requiera asignar un marco a un segmento determinado y no lograra encontrar uno disponible deberá intercambiar a disco alguno de los que estén en uso para poder liberarlo y luego asignarlo a dicho segmento.

Debería decir:

Cuando la MSP requiera asignar un marco a una página determinada y no lograra encontrar uno disponible, deberá intercambiar a disco alguno de los que estén en uso para poder liberarlo y luego asignarlo a dicha página.

- Pagina 17:

La MSP soportará una cantidad máxima de segmentos por programa de 4096, tamaño de los mismos de hasta 4096 Bytes, páginas de tamaño fijo 256 Bytes.

El tamaño máximo de los segmentos no es 4096 bytes, sino 1048576 (4096 \* 256) bytes.

- Página 18:

Donde dice:

devuelve la cantidad en bytes Tamaño comenzando

Deberia decir:

devuelve hasta Tamaño bytes comenzando

- Página 12:

Los parámetros del archivo de configuración del Kernel son: Puerto de Escucha, Dirección IP de la MSP, Puerto de la MSP, Tamaño del Quantum, Ubicación Syscalls

Debería agregarse, además, el parámetro Tamaño del Stack.

- Página 7:

Campo: Base del Segmento de Código | Tipode dato: Numérico

La base del segmento de código debería ser de tipo "Dirección"

- Página 14:

Cada vez que el CPU quiera ejecutar una instrucción, realizará los siguientes pasos:

1. Cargar los registros de la CPU con los datos del TCB a ejecutar.

La carga de los registros se deberá hacer sólo al recibir el TCB, y *no* en cada ciclo de instrucción.

- Páginas 24 y 25:

El registro de flags **no deberá** implementarse. La operación `DIVR` deberá abortar con un fallo de división por cero en lugar de setear el flag de `ZERO_DIV`, y la operación `FLCL` queda deprecada y no deberá implementarse.

- Si bien los conceptos de **dirección lógica** y **dirección virtual** son distintos (ver la sección 2.1 de [Understanding the Linux Kernel](#) y la sección 4.8.7 de [este otro documento](#)), a lo largo del enunciado (y en algunos otros documentos, como el Operating Systems Concepts de Silberschatz) ambos términos se usan indistintamente para referirse al mismo concepto.

