Este documento detalla las instrucciones y el flujo que genera su ejecución dentro del programa, a partir del momento que la CPU la ejecuta.

Las instrucciones están en un archivo mCod que se manda ejecutar.

Las instrucciones son:

[iniciar N](#INST_INICIAR)

[leer N](#INST_LEER)

[escribir N “texto”](#INST_WRITE)

[entrada-salida T](#INST_IO)

[finalizar](#INST_END)

**Instrucción: iniciar N**

Reacción del hilo CPU que lo ejecuta:

* Me comunico con el AdM informándole en número de páginas necesarias.
* Escucho a la memoria, que me avisará si hubo un problema con la cantidad de páginas pedidas.
* En ese caso termino la ejecución de ese programa.
* Guardo en un buffer la expresión detallada en Anexo I.

Reacción del AdM:

* Escucho un mensaje de un CPU pidiéndome N páginas.
* Asigno la memoria para ese nuevo proceso:
  + Creo una tabla de páginas para ese proceso.
  + Asigno en una lista esas N páginas
* Me comunico con el Swap para que reserve N páginas.
* Escucho la respuesta del Swap a ver si rechazó o aceptó el proceso.
* Si Swap rechazó el proceso, le aviso a CPU que no se puede ejecutar eso.

Reacción del Swap:

* Escucho al AdM pidiéndome reservar N páginas.
* Me fijo si tengo lugar TOTAL para meter las N páginas.
* Si no lo tengo, devuelvo un mensaje de error, rechazando el proceso.
* Busco un hueco libre de N páginas.
* Si no lo encuentro, sé que hay lugar y procedo a la compactación:
  + Compacto.
  + Genero una espera según cómo esté yo configurado.
* Finalmente, reservo el hueco que debería haberse creado ya.

**Instrucción: leer N**

Reacción del hilo CPU que lo ejecuta:

* Me comunico con el AdM informándole la página que quiero.
* Escucho a la memoria, que me avisará si hubo un problema.
* En ese caso termino la ejecución de ese programa.
* Caso contrario, me devolverá la página que necesito.
* Guardo en un buffer la expresión detallada en Anexo I.

Reacción del AdM:

* Escucho un mensaje de un CPU pidiéndome una página.
* Traduzco a Marco el número de página pedido:
  + Busco en la TLB a ver si ya existe esa traducción.
  + Si no existe, la busco en la tabla de páginas
* Con la dirección real ya en mi poder, busco en memoria la página.
* Si no encuentro la página, se la pido al Swap.
  + Recibo la página desde Swap
  + Procedo a correr un método de carga
  + Envío a Swap la página que desalojé.
* Leo la página que ahora estará en la memoria.
* Devuelvo al hilo CPU el contenido de esa página.

Reacción del Swap:

* Escucho al AdM pidiéndome una página.
* La busco en mi archivo Swap.
* Genero una demora según mi configuración.
* Le devuelvo al AdM esa página.
* Espero que el AdM me dé la página que desalojó.
* La guardo en lugar de la que le dí.

**Instrucción: escribir N “texto”**

Reacción del hilo CPU que lo ejecuta:

* Me comunico con el AdM informándole la página que quiero, junto con la cadena “texto” a escribir.
* Escucho a la memoria, que me avisará si hubo un problema.
* En ese caso termino la ejecución de ese programa.
* Guardo en un buffer la expresión detallada en Anexo I.

Reacción del AdM:

* Escucho un mensaje de un CPU pidiéndome una escritura de página y mandándome una cadena.
* Traduzco a Marco el número de página pedido:
  + Busco en la TLB a ver si ya existe esa traducción.
  + Si no existe, la busco en la tabla de páginas
* Con la dirección real ya en mi poder, busco en memoria la página.
* Si no encuentro la página, se la pido al Swap.
  + Recibo la página desde Swap
  + Procedo a correr un método de carga
  + Envío a Swap la página que desalojé.
* Escribo “texto” en la página que ahora estará en la memoria, rellenando el resto con ceros.
* Devuelvo al Swap la página de ser necesario el contenido de esa página. REVISAR ESTO.
* Devuelvo si quiero la página al hilo CPU.

Escuchar en mi flujo de entrada de CPU el pedido de escritura de una página, junto con el número de página y su contenido.

Traducir a Marco ese numero de pagina recibido.

Buscar en mi TLB a ver si ya está traducida. Caso contrario lo busco en la tabla de páginas.

Buscar en memoria esa pagina.

Si no esta, pedirsela al swap y ejecutar algoritmo de carga.

Escribir la pagina con el contenido recibido.

(si se quiere, devolver contenido).

Reacción del Swap: REVISAR

* Escucho al AdM pidiéndome una página.
* La busco en mi archivo Swap.
* Genero una demora según mi configuración.
* Le devuelvo al AdM esa página.
* Espero que el AdM me dé la página que desalojó.
* La guardo en lugar de la que le dí.

Sobreescribir la página con lo que recibí.

**Instrucción: entrada-salida T**

Reacción del hilo CPU que lo ejecuta:

* Me comunico con el Planificador informándole que bloquee este proceso por T segundos.
* Guardo en un buffer la expresión detallada en Anexo I.
* Devuelvo el buffer completo al Planificador.
* Informo al Planificador que quedé libre para ejecutar otras cosas.

Reacción del AdM:

* N/A

Reacción del Swap: REVISAR

* N/A

**Instrucción: finalizar**

Reacción del hilo CPU que lo ejecuta:

* Me comunico con el AdM informándole que el programa terminó.
* Guardo en un buffer la expresión detallada en Anexo I.
* Devuelvo el buffer completo al Planificador.
* Informo al Planificador que quedé libre para ejecutar otras cosas.

Reacción del AdM:

* Escucho un mensaje de un CPU avisándome del fin del programa.
* Liberar la memoria asignada a ese proceso.
* Destruir su tabla de páginas.
* Limpiar la TLB de esas páginas.
* Me comunico con Swap avisándole que el programa terminó.

Reacción del Swap:

* Escucho un mensaje del AdM avisándome del fin del programa.
* Liberar las estructuras del archivo de páginas Swap correspondientes a ese proceso.