

Sistemas Operativos II - LCC  
Práctica 4 - 2013.  
Memoria Virtual  
*Entrega:* 26 de Junio.

**NOTA:** Debe utilizar el SVN de la materia creando un subdirectorio por alumno/grupo en:  
<https://dcc.fceia.unr.edu.ar/svn-no-anon/lcc/R-412/Alumnos/2013/>

## 1. Memoria virtual y TLB

Para esta práctica no hay nuevos archivos, sin embargo de acuerdo a la constante `USE_TLB` definida en el Makefile del nuevo directorio de trabajo (vm) se comienza a utilizar la TLB en lugar de las tablas de paginación manejadas por el procesador. Los programas `matmult` y `sort` son dos ejemplos de programas que hacen uso extensivo de la memoria virtual y se recomienda utilizarlos para probar el correcto funcionamiento de las implementaciones (además crear otros propios).

Algunos ejercicios requieren modificar clases del directorio `machine` (para obtener estadísticas por ejemplo). Se debe tener cuidado de no alterar la funcionalidad de esas clases.

1. Implementar TLB. Detalles: cada espacio de direcciones tiene una tabla de páginas que ya no se actualiza automáticamente con el hardware. La única asistencia del hardware ahora es que cuando una entrada no figura en la TLB se dispara un `PageFaultException`, además si la entrada figura en la TLB pero es de sólo lectura se genera una `ReadOnlyException`. Se debe implementar una rutina para solucionar estos inconvenientes (para probar los programas de ejemplo se debe ampliar el tamaño de la memoria física).
2. Calcular el porcentaje de aciertos (hit ratio) de la TLB (al menos para dos programas razonablemente grandes). Ver cómo se modifica este porcentaje al agregar más entradas a la TLB (32 entradas = MIPS R2000, 64 = MIPS R4000). ¿Qué tamaño sugeriría para la TLB?

3. Implementar carga por demanda (“demand loading”) pura. Suponer por ahora que los programas entran en memoria (agrandarla de no ser así). Para esto el constructor del espacio de direcciones no debe leer ninguna página de código o datos del ejecutable sino que estas se cargarán cuando se requiera acceder a esa página (pagefault).
4. Implementar paginación utilizando la política más sencilla posible (FIFO o algo similar). Ahora se necesita un archivo para guardar las páginas que no entren en memoria y de esa forma proveer la ilusión de una memoria mucho más grande que la física. Crear un archivo con nombre SWAP.asid para cada espacio de direcciones, en donde la página n del espacio de direcciones se pueda guardar en el bloque n del archivo. Se debe mantener un mapa de la memoria (o coremap) que indique a qué página virtual de qué proceso corresponde una página física. Probar la implementación ejecutando un solo proceso, luego con procesos sucesivos y luego con procesos concurrentes (opción -rs).
5. Mejorar la política de paginación (se recomienda ir introduciendo mejoras sucesivas). Implementar LRU o el algoritmo mejorado del reloj (o alguno que considere similar).
6. **Opcional(pero interesante)** Comparar el rendimiento (cantidad de accesos a disco, fallos de páginas totales, ...) entre los dos puntos anteriores y también contra el algoritmo óptimo (hacerlo al menos con dos programas).