Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2343 – Arquitectura de Computadores (I/2018)

Enunciado Tarea 02

Memoria virtual y paginación

Fecha de entrega: Mieercoles 9 de Mayo del 2018, 23:59 hrs.

1. Descripción

En esta tarea tendrán la oportunidad de experimentar con el diseño de un sistema de multiprogramación, desde el punto de vista de la memoria y de la CPU. Más específicamente, deberán programar la simulación de una memoria virtual que incluye una *Translation Lookaside Buffer* y de un algoritmo de *scheduling*, cuando se ven enfrentados a la ejecución de diversos programas que realizan peticiones a memoria.

2. Input

La tarea desarrollada debe recibir como *input* las especificación completa de la memoria virtual junto con las instrucciones realizadas por cada programa. Se subió en el Syllabus el código necesario para leer los *inputs*

Los *inputs* vendrán en el siguiente orden

- MEM FIS SIZE: Indica el tamaño en bytes de la memoria física de datos.
- MEM VIR SIZE: Indica el tamaño en bytes de la memoria virtual de datos.
- PAGE SIZE: Indica el tamaño en bytes de cada página.
- SUBSTITUTION: Indica la política de sustitución de páginas en la memoria física. Esta puede ser "FIFO", "LRU" o "LFU".
- NUM LINE: Indica el número de líneas de la memoria caché.
- CORR: Indica la función de correspondencia a utilizar en la memoria caché. Esta puede ser "DM" (*directly mapped*) o "FA" (*fully associative*).
- SUBS: Indica la política de sustitución a utilizar en la caché. Esta puede ser "FIFO", "LRU" o "LFU".
- NUM PROG: Indica el número de programas a ejecutar.
- PROG: Una lista de lista, donde cada lista de adentro representa un programa, el cual indica una secuencia ordenada de largo arbitrario para acceder a datos de memoria. El primer elemento de la lista representa el programa 1, el segundo elemento al programa 2 y así sucesivamente.

A continuación se muestra el código necesario para leer las especificaciones de la caché y memoria virtual.

```
# Este código debe estar al inicio de su programa
import json

MEM_FIS_SIZE = int(input())
MEM_VIR_SIZE = int(input())
PAGE_SIZE = int(input())
SUBSTITUTION = input()
NUM_LINE = int(input())
CORR = input()
SUBS = input()
NUM_PROG = int(input())
PROG = json.loads(input())
```

Un ejemplo de los inputs usando el código anterior, será:

```
512
1024
32
"FA"
4
"FA"
"FIFO"
4
"[[1,2,4,6,2,-1,4,7,3,5,-1,2,1],[6,4,2,5,7,8,5,4,5,-1,5],[1,4,5,6,7],[1,2,3,4,5,6,3]]"
```

Puede tomar las siguientes consideraciones:

- Existe una palabra por linea en la caché.
- Las tablas de páginas no utilizan espacio en memoria.
- El cambio de contexto (*scheduling*) será cooperativo, en donde cada ***yield*** que será representado con los accesos a memoria "-1". Por ejemplo, PROG 1 tiene dos ***yield*** mientras que PROG 2 tiene un ***yield*** y los otros dos programas no tienen ninguno.
- Cada palabra de la TLB es una entrada de página.
- Los procesos no tienen páginas asignadas cuando empiezan a ejecutarse (todas son inválidas).
- Solo se considera HIT en la caché cuando la página a la que se accede es válida.
- La capacidad de almacenamiento del *swap file* es infinita.
- El orden de ejecución de los programas está dado por el número del programa.
- El *input* solo serán valores definidos anteriormente. En ningún momento se ingresarán números o datos que no corresponden a la propiedad definida.
- Todos los tamaños y número de palabras o bloques serán potencias de 2 mayores a 1.

3. Output

Como salida, la tarea debe imprimir **exactamente** los siguientes valores por programa:

- Número de *hits* a la TLB. El formato es: "hit TLB: XX" donde XX es el número de *hits*.
- Número de *page faults*. El formato es: "page fault: XX" donde XX es el número de *page fault*.
- Número de *swap in*. El formato es: "swap in: XX" donde XX es el número de *swap in*.
- Número de *swap out*. El formato es: "swap in: XX" donde XX es el número de *swap out*.
- Número de páginas válidas. El formato es: "page valid: XX" donde XX es el número de páginas válidas.
- Número de páginas en disco. El formato es: "page disk: XX" donde XX es el número de páginas en disco.

Un ejemplo del formato de *output* para 2 programas:

```
PROGRAMA 1
hit TLB: 5
page fault: 4
swap in: 5
swap out: 2
page valid: 3
page disk: 1

PROGRAMA 2
hit TLB: 0
page fault: 2
swap in: 1
swap out: 2
page valid: 3
page disk: 1
```

Debe tomar las siguientes consideraciones al momento de entregar el *output*.

- Debe imprimir los datos en el mismo orden del pdf y con el formato solicitado (incluir un salto de línea entre el un programa y otro).
- No debe imprimir nada durante la ejecución del programa.

4. Implementación

Para la realización de esta tarea deberán trabajar en el lenguaje Python 3.x, sin ocupar bibliotecas externas que implementan la lógica de la tarea. Está permitido el uso de bibliotecas para leer archivos, imprimir en pantalla y manejar parámetros por línea de comando. Para el uso de alguna librería que no sea para lo indicado anteriormente, deben hacer una *issue* donde pregunten por la librería y se explique el motivo de su uso.

5. Corrección

Para ejecutar la tarea y corregirla, esta deberá recibir inputs en el mismo orden que se especificó en el enunciado.**Utilicen el código entregado en el Syllabus**.

Luego de esto, el programa deberá imprimir en pantalla los resultados requeridos. No se evaluará código en esta ocasión, solo que el *output* sea el indicado y que no se use ninguna librería prohibida.

6. Contacto

- Francesca Lucchini: flucchini@uc.cl
- Felipe Pezoa: fipezoa@uc.cl
- Hernán Valdivieso: hfvaldivieso@uc.cl
- Luis Leiva: lileiva@uc.cl

7. Integridad académica

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno, sin apoyo en material de terceros. Por "trabajo" se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros. Si un alumno copia un trabajo, obtendrá nota final 1,1 en el curso y se solicitará a la Dirección de Docencia de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral. Por "copia" se entiende incluir en el trabajo presentado como propio partes hechas por otra persona.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile. Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.