

## **Sistemas Operativos B**

### **Práctica 2**

#### **Objetivo:**

El objetivo de esta práctica es implementar la paginación por demanda pura en Nachos. Es decir, el proceso empieza su ejecución con cero páginas cargadas en memoria y se van a ir cargando las páginas que van siendo referenciadas.

#### **Puntos a implementar:**

1. Inicializar la tabla de páginas con el bit de validez en un valor de inválido para cada entrada. La tabla de páginas debe tener un tamaño igual a la cantidad de marcos que se tienen en la arquitectura, es decir, si tengo definidos 32 marcos definidos la tabla de páginas será de 32 entradas; si tengo 16 marcos, la tabla de páginas tendrá 16 entradas, y así respectivamente.
2. Crear el archivo de intercambio del proceso a ejecutar (práctica anterior).
3. Agregar la excepción de fallo de página al manejador de excepciones.
4. Cada que se genere un fallo se deberá seguir el procedimiento para manejar el fallo de página. Se debe tener mucho cuidado en el cálculo de la dirección dentro del archivo para obtener la página deseada y hacer correctamente el intercambio hacia adentro (swap in).  
  
Se deberá asignar en orden ascendente los marcos a la páginas, para la primer página que genere fallo se le asignará el marco 0, para la segunda página que genere fallo se le asignará el marco 1, para la tercer página que genere fallo se le asignará el marco 2 y así sucesivamente.
5. Imprimir en pantalla cada referencia a memoria (la dirección lógica, vpn, offset y la dirección física) y el número de fallo que genera (si es que lo hay).
6. Además se deberá de indicar la cantidad total de fallos de página en el resumen de salida cuando el programa termina correctamente (como se muestra para halt)

Machine halting!

Cadena de referencia: .

Ticks: total 23, idle 0, system 10, user 13

Disk I/O: reads 3, writes 0

Console I/O: reads 0, writes 0

Paging: faults 3

Network I/O: packets received 0, sent 0

Cleaning up...

## NOTAS IMPORTANTES

- El archivo halt debe ejecutarse completo y mostrar la misma salida que en la práctica 0, la única **diferencia es el contenido de la tabla de páginas**. Es decir, el orden en que se asignaron los marcos va ser diferente, pero las direcciones lógicas a las que se hace referencia deben ser iguales a las de la práctica 0.
- Realice la implementación de práctica de forma general, es decir, que al cambiar el tamaño de la página o cambiar la cantidad de marcos no afecte en la prueba de la práctica y no tengan que modificar el código. Para esto realice sus condicionales, comparaciones, etc. utilizando las constantes definidas en nachos.
- Considerando la versión original de Nachos, el archivo matmult y sort serán ejecutados parcialmente, es decir, se van a ejecutar hasta que se hayan cargado las primeras 32 páginas diferentes que requiera el proceso y posteriormente terminará la ejecución por error de que ya no existe espacio en memoria.
- Se deberá de probar cambiando la cantidad de marcos (que sea una cantidad mayor y menor) definidos en la arquitectura y dependiendo de este parámetro serán las páginas que deberán cargarse en memoria. Es importante que entienda que los programas se ejecutarán parcialmente en el caso de que no haya suficiente espacio en memoria para cargar todo el proceso.