Memory Management

A estas alturas ya debería estar familiarizado con el funcionamiento interno de PintOS.

Recuerde que **PintOS** debe manejar correctamente varios threads en ejecución, sincronizados y debería poder cargar varios programas. Actualmente estará limitado por el tamaño de la memoria principal, acotando la cantidad o tamaño de los programas ejecutados, por lo que el manejo de memoria es de vital importancia para un sistema operativo.

 En el laboratorio se realizará una pequeña simulación que consiste en implementar una administración eficiente de la memoria disponible y llevar el control de la PageTable de cada proceso (Fase 2). Su implementación debe contemplar lo siguiente:

La memoria física que utilizaremos puede almacenar hasta 64 páginas.

Cada **proceso** tienen ciertas características al crearse.

Deben especificar el **nombre** (sequencial) y la **cantidad de páginas** que ocupa (rango)

Debe tener un pid único y su PageTable para el direccionamiento de la memoria.

La **PageTable** tiene la siguientes identificadores:

pid: Process ID

virtualPage: Número(s) de página(s) virtual(es) physicalPage: Número(s) de página(s) física(es)

En PintOS, una **PageTable** es una estructura de datos que la CPU utiliza para traducir una dirección virtual a una dirección física, es decir, de una page a un frame. El formato de la PageTable es dictado por la arquitectura 80x86 (Read: 4.1.2 Memory Terminology).

Al terminar el proceso debe liberar las páginas que este ocupaba en memoria y así puedan ser utilizadas por otro proceso.

Cada proceso debe <u>"durar o ejecutarse"</u> durante un tiempo aleatorio mayor a 5 segundos y menor que infinito.

Importante:

- Los procesos serán threads que se deben generar automáticamente con un espacio de tiempo aleatorio (rango) entre la creación de uno y la creación del siguiente.
- Por lo que debe asignar eficientemente las páginas disponibles en memoria, esto quiere decir que debe ser capaz de asignar páginas libres aunque estas no estén contiguas.
- Si un proceso contiene una cantidad de páginas mayor a las páginas disponibles en memoria debe indicar al usuario un warning que dicho proceso ya no puede ser cargado debido a memoria insuficiente. (Extra en este mismo laboratorio, leer extra puntos)
- Al cargar un proceso a memoria, debe desplegar el contenido de la PageTable del proceso y
 el estado actual de la memoria indicando que páginas están ocupadas y que páginas están
 libres de manera que sea fácilmente legible y entendible. (Se recomienda usar una matriz de
 8x8 para representar la memoria física disponible con CERO y no disponible con UNO, la
 forma de mostrar la PageTable queda a su discreción)

```
[xx][0][1][2][3][4][5][6][7]
[0x][0][0][0][0][0][0][0][0]
[1x][0][0][0][0][0][0][0][0]
[2x][0][0][1][1][1][0][0][0]
[3x][0][0][1][1][1][0][0][0]
[4x][0][0][1][1][1][0][0][0]
[5x][0][0][0][0][0][0][0][0]
[6x][0][0][0][0][1][1][1][1][1][1]
```

[7x][0][0][0][1][1][1][1][1]

Entregable:

Este laboratorio debe ser realizado en C junto a un archivo MAKE para facilitar la carga de cualquier archivo externo o ejecución, cualquier detección de copia de calificará con -100 y de no compilar se calificará con 0. Este laboratorio tiene una calificación de 100% y si realiza exitosamente lo extra se le calificará doble.

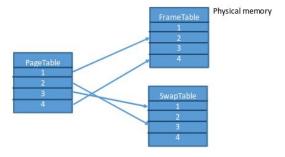
Extra Puntos (+100%):

Si se percató en el laboratorio base la PageTable sólo contempla la memoria física como límite y muestra al usuario un **warning** que el proceso ya no puede ser cargado debido a <u>memoria insuficiente</u>, PintOS implementa el Swapping (Fase 3), en el laboratorio **debe simular ese control** para hacer creer a los procesos que hay espacio.

Se agregará otro identificador a la PageTable **swp** que indicará si es dirección de memoria o disco, no se realizará como tal un algoritmo de reemplazo (use, valid, dirty, lock) ya que los thread se estarán ejecutando un tiempo determinado por lo que supondremos un FIFO entre Memoria Física y Disco.

Se conserva la información que se despliega en el laboratorio, cada proceso al cargarse debe imprimir su PageTable y ahora el estado de los <u>dos espacios de memoria</u>. El proceso que <u>anteriormente levantaba el warning</u> se asignará en la memoria asignada en disco **pero no se ejecutará y espera** hasta que existan las páginas necesarias para trasladarlo a memoria física.

Al momento que un proceso en ejecución en Memoria Física termine y libere las páginas que ocupa se realizará el <u>Algoritmo de Selección de Proceso Entrante</u> el cual queda a su discreción, este se encargará del paso de un nuevo proceso o que esté en disco hacia la memoria física. Para el laboratorio no existe la FrameTable y la SwapTable como tal, solo la representación de memoria física y disco, recuerde que en PintOS si existen.



La direcciones de memoria asignada en disco al Swapping será la siguiente **con páginas ya utilizadas (1)** por otros archivos que **no podrán ser usados**, teniendo 90 páginas disponibles.