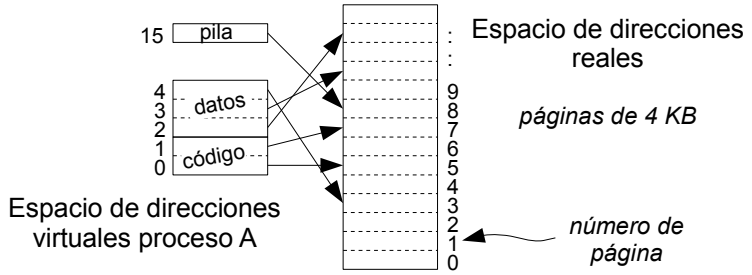
CC4302 Sistemas Operativos
Control 3– Semestre Otoño 2012
Prof.: Luis Mateu

Pregunta 1

Parte i.- La siguiente figura muestra la asignación de páginas de un proceso A que se ejecuta en un sistema Unix.



Haga la tabla de páginas para el proceso A indicando en la tabla: número de página virtual, número de página real y atributos de validez y escritura (V y W).

Parte ii.- Suponga que el proceso A de la parte i.- invoca **fork**. Haga las tablas de páginas (especificando V y W) para el hijo y el padre justo después del **fork** considerando que se usa la técnica *copy-on-write*.

Parte iii.- Se necesita agregar a Unix la llamada al sistema `migrate(host)` que hace que el proceso que la llama migre a otro computador cuya identificación es `host`. Es decir al retornar el proceso continúa su ejecución en el nuevo `host`.

- Explique brevemente cómo implementaría esta llamada al sistema en una máquina con *segmentación*.
- Explique si es o no posible implementar esta llamada en un sistema multiproceso que no posee una MMU (memory management unit).

Parte iv.- Compare las dos estrategias de *paginamiento en demanda* vistas en el curso desde el punto de vista de (a) sobrecosto en tiempo de ejecución cuando la memoria física sobra, (b) *page faults* cuando hay penuria de memoria pero hay un solo proceso en ejecución, (c) *page faults* cuando hay penuria de memoria y hay muchos procesos en ejecución.

Pregunta 2

Parte a.- En una aplicación se requiere implementar un diccionario. Se consideran 2 implementaciones:

```
typedef struct { char key[8]; char data[24]; } Entry;
```

`Entry dict[1000];`	`Entry *dict[1000];`

En la primera implementación toda la información se encuentra contigua en memoria. En la segunda, las entradas del diccionario pueden quedar muy dispersas y desordenadas en la memoria debido a que el heap que maneja malloc está fragmentado. Suponga que las búsquedas en el diccionario son secuenciales. Estime para ambas implementaciones el peor caso del número de fallas en la TLB (*Translation Lookaside Buffer*) al hacer una búsqueda. ¿Cuantos accesos adicionales a la memoria significaría cada falla en la TLB considerando un microprocesador Intel x86?

Parte b.- Suponga que en un sistema paginado la MMU no implementa el bit R (referencia). Reescriba para este sistema el pseudo código de la estrategia del reloj visto en clase de cátedra. Ud. puede lograr la funcionalidad del bit R usando astutamente el bit V (validez) y un bit adicional en la tabla de páginas que es manejado completamente en software por el sistema operativo (ignorado por el hardware). En su pseudo código indique claramente que se debe hacer en caso de *page-fault*.

Parte c.- El siguiente diagrama muestra con una r las lecturas en memoria y con una w las escrituras para un proceso que se ejecuta en un sistema Unix que utiliza la estrategia del *working set*.

página	A	B	C	D	E	F	G	*Tiempo*
6		r	r r	w w	r	w		
5	r w	r		r r r		r	r	
4		r					r	
3	r		r r r		w w w	ww		
2	r r r	r	r	r r	w r	w w	r	
1	r r	r r	r	r w				
0		w w	r		r	w		

La letras A, B, C, etc. denotan los intervalos para los que se calcula el working set. Los números 0, 1, 2, etc. denotan las páginas del proceso. Suponga que al inicio de A todas la páginas tienen el atributo *Dirty* en falso. Conteste:

i. Indique para los períodos C a F qué accesos pueden producir *page-faults*. Utilice coordenadas del estilo *(G, 4, 1er. acceso)*.
ii. Indique el valor del atributo *Referenced* para todas las páginas al inicio del intervalo E y al final de ese intervalo.
iii. Suponga que al inicio de E el atributo *Dirty* de la página 5 es falso. Explique si el acceso (D, 5, 1er. acceso) produjo o no un *page-fault*.
iv. Indique el valor del atributo *Dirty* para cada página al inicio del intervalo E y al final de ese intervalo, suponiendo que no hubo ningún *page-fault* en el período A-E.