

SISTEMAS OPERATIVOS

EXAMEN

Pauta de Corrección

Wenceslao Palma <wenceslao.palma@pucv.cl>

1. Responda las sgtes preguntas. Justifique cada una de sus respuestas.

- (5 ptos) Si s es un semáforo, qué sucede cuando un proceso ejecuta la primitiva $wait(s)$.
R.: Se decrementa el contador asociado al semáforo, si luego de eso el valor es mayor o igual que cero el proceso ingresa en la sección crítica, en caso contrario el proceso debe esperar su turno para acceder a la sección crítica por orden de llegada.
- (5 ptos) Por qué un ordenamiento lineal de los recursos previene el deadlock?
R.: Porque evita la espera circular asociando un índice a cada recurso y solicitando acceso en orden creciente. El recurso R_i antecede al recurso R_j si $i < j$, de esta manera si dos procesos P_1 y P_2 están en deadlock se debe a que P_1 adquirió R_i y espera por R_j mientras que P_2 adquirió R_j y espera por R_i . Esto último es imposible ya que implica que $i < j$ y al mismo tiempo $j < i$.
- (5 ptos)Cuál es la diferencia entre prevención, predicción y detección de deadlock?
R.: Existen cuatro condiciones que se deben cumplir simultáneamente para que se produzca deadlock: **1** Exclusión mutua, **2** Retención y espera, **3** No apropiación y **4** Espera circular. La prevención evita la aparición de alguna de las tres primeras, con la predicción es posible alcanzar las 3 primeras condiciones pero se toman acciones para que no se produzca deadlock, y con la detección el deadlock ocurre ya que no se limita el acceso a los recursos y una vez que ocurre deadlock se toman acciones para eliminar la espera circular.
- (5 ptos) Al realizar una llamada al sistema ocurre un cambio de contexto? Si su respuesta es afirmativa, indique en que estado queda el proceso que abandona el procesador.
R.: Sí, una llamada al sistema es una invocación que realiza un proceso para obtener un servicio otorgado por el kernel, luego debe existir un cambio de contexto para que la rutina del kernel se ejecute. En este caso el proceso que realiza la llamada al sistema pasa al estado bloqueado.

2. (20 ptos.) Considere un sistema con memoria virtual donde el conjunto residente es de tamaño 5. Muestre el funcionamiento de los algoritmos de reemplazo de página LRU y Clock para un proceso que realiza las siguientes referencias a páginas :

7 3 15 7 17 8 13 6 15 7 11 5 15 7 11 5 9 5 9 5.

LRU (10 ptos.)

7	3	15	7	17	8	13	6	15	7	11	5	15	7	11	5	9	5	9	5
7	7	7	7	7	7	7	7	15	15	15	15	15	15	15	15	15	15	15	15
	3	3	3	3	3	13	13	13	13	13	5	5	5	5	5	5	5	5	5
		15	15	15	15	15	6	6	6	6	6	6	6	6	6	9	9	9	9
			17	17	17	17	17	7	7	7	7	7	7	7	7	7	7	7	7
					8	8	8	8	8	11	11	11	11	11	11	11	11	11	11
						F	F	F	F	F	F								F

CLOCK (10 ptos.)

7	3	15	7	17	8	13	6	15	7	11	5	15	7	11	5	9	5	9	5
*7-1	*7-1	*7-1	*7-1	*7-1	*7-1	13-1	13-1	13-1	13-1	*13-1	13-0	15-1	15-1	15-1	15-1	15-1	15-1	15-1	15-1
	3-1	3-1	3-1	3-1	3-1	*3-0	6-1	6-1	6-1	6-1	6-0	*6-0	*6-0	*6-0	*6-0	9-1	9-1	9-1	9-1
		15-1	15-1	15-1	15-1	15-0	*15-0	*15-1	15-0	15-0	5-1	5-1	5-1	5-1	5-1	*5-1	*5-1	*5-1	*5-1
			17-1	17-1	17-0	17-0	17-0	7-1	7-1	*7-1	7-0	7-1	7-1	7-1	7-1	7-1	7-1	7-1	7-1
					8-1	8-0	8-0	8-0	*8-0	11-1	11-1	11-0	11-0	11-1	11-1	11-1	11-1	11-1	11-1
						F	F	F	F	F	F	F	F						F

3. (20 ptos.) Escriba un script usando bash que busque, a partir del directorio actual, todos los archivos en formato pdf y los transforme a formato ps (**\$ pstopdf tarea.pdf**, transforma **tarea.pdf** en **tarea.ps**). El script debe entregar como resultado la cantidad de archivos transformados y su ubicación dentro del árbol de directorios.

```
#!/usr/bin/bash

transformar () {
  for archivo in `ls`; do
    if [ -d "$archivo" ]; then
      cd "$archivo"
      transformar
      cd ..
    else
      if echo $archivo | grep -qi "\.pdf"; then
        echo "transformando.....$PWD/$archivo">>$salida
        pstopdf $archivo
        i=`expr $i + 1`
      fi
    fi
  done
}
dir=$PWD
salida="$dir/out.txt"
i=0
transformar
echo "nro de archivos eliminados: $i">>$salida
```