

# SISTEMAS OPERATIVOS

## EXAMEN

### Pauta de corrección

Wenceslao Palma <wenceslao.palma@ucv.cl>

1. Responda las sgtes preguntas. Justifique cada una de sus respuestas.
  - (a) (10 ptos.) Considere un espacio de direcciones lógicas paginado, compuesto de 32 páginas de 2KB, correspondiente a un espacio de memoria física de 1MB. Cuál es el formato de las direcciones lógicas y físicas de los procesos? Cuál es el efecto sobre la tabla de páginas si el espacio de memoria física se reduce a la mitad?  
R.: Si tenemos 32 páginas se necesitan 5 bits para representarlas. Luego el máximo desplazamiento dentro de una página equivale a  $2^{11}$  bytes (2KB). Por lo tanto la dirección lógica es de 16 bits compuesta de 5 bits para la página y 11 bits para el desplazamiento. Ya que se tiene un espacio de 1MB para direccionar, cada entrada en la tabla de páginas necesita 9 bits para determinar la dirección del frame. Si la memoria se reduce a la mitad se necesitarán 8 bits para la dirección del frame.
  - (b) (5 ptos.) Considerando la prevención de deadlock explique cómo se puede prevenir la espera circular.  
R.: Se puede prevenir definiendo un orden lineal sobre los recursos. Para esto se asocia un índice a cada recurso. El recurso  $R_i$  antecede a  $R_j$  en la ordenación si  $i < j$ . Si dos procesos  $A$  y  $B$  están interbloqueados es porque  $A$  adquirió  $R_i$  y solicitó  $R_j$ , mientras que  $B$  ha adquirido  $R_j$  y solicitado  $R_i$ . Esto es imposible porque implica que  $i < j$  y  $j < i$ .
  - (c) (5 ptos.) Cuál es el número mínimo de procesos y recursos necesarios para que se produzca deadlock?  
R.: Para que se produzca interbloqueo se necesitan como mínimo dos procesos y dos recursos. De este modo se puede configurar un ciclo de retención y espera que lleva a la aparición de deadlock.

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 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
  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  7
        8  8  8  8  8  11  11  11  11  11  11  11  11  11  11  11
          F  F  F  F  F  F  F  F  F  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
  3-1  3-1  3-1  3-1  3-1 *3-0 6-1  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 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 11-1 11-1
          F  F  F  F  F  F  F  F  F  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

```