

EXAMEN SISTEMAS OPERATIVOS AVANZADOS

TEORIA
FEBRERO 2018

PREGUNTA 1 PROCESOS (2.5 PUNTOS)

- Describir como es el procedimiento que se emplea (p.ej. en xv6) para leer desde el código de la llamada al sistema sus argumentos de entrada.
- Describir cuando se regresa al *idle loop* del planificador (por ejemplo, el proceso en ejecución realiza una operación de I/O). ¿Cómo se identifica la instrucción que tiene que ejecutar a continuación?
- ¿Cual es objetivo fundamental de los algoritmos de planificación de múltiples colas con *feedback*?
- ¿Qué problemas tienen los algoritmos de planificación de sistemas multiprocesador basados en una sola cola (como por ejemplo SQMS)?
- ¿Por qué, habitualmente, el quantum de tiempo en un planificador Round Robin en un sistema operativo de escritorio debería ser más pequeño que el de uno de un servidor? En qué cargas de trabajo crees que el del servidor sería similar al del equipo de escritorio.

PREGUNTA 2 MEMORIA (2.5 PUNTOS)

- Escribir un fragmento de código que permita saber la VA del segmento de código, del *stack* y del *heap* del proceso.
- Suponer que la *free-list* del *heap* de un proceso tiene únicamente disponible 4KB. ¿Qué ocurre si realizamos un `malloc()` para almacenar un array de 2048 enteros de 32-bits?
- Imaginar un caso en el que la PTE contiene solo contiene el número de marco y un bit de validez (no incluye ni bit de presencia, bit de dirty, etc...). Este bit lo tiene **situado en el menos significativo de la entrada**. Suponer que tenemos un espacio de direcciones virtual de **128 bytes** con páginas de **32 bytes**. El contenido de la tabla de páginas es: **0xFE, 0x0F, 0xFE, 0x09**. ¿Qué dirección física se accederá en cada una de las siguientes direcciones virtuales?
 - 0x065
 - 0x00c
 - 0x026
 - 0x058
- Suponer una tabla de páginas en dos niveles. Suponer un espacio de direcciones virtual de **15-bits** con páginas de **32bytes**. El formato de PDE y PTE es el mismo: un bit de validez seguido de **7** bits para referenciar el marco de página. El PDBR vale **73** en decimal. A continuación, se muestra un volcado parcial de la memoria:

```
0x 6: 0a 1c 01 14 0b 1a 19 0a 0a 1a 0c 14 02 0c 1c 0c 15 04 0e 13 17 11 08 05 08 07 04 13 0f 1d 0f 1e
0x 73: a2 d2 97 96 d9 7f 87 b4 b7 f2 f4 82 bf 7f be 93 e8 9d 99 9e f1 7f 7f b0 d8 da eb b1 81 c3 c2 f6
0x 114: 7f 7f 7f 82 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
```

Se pretende acceder a la dirección virtual 0x1787. ¿Qué traducción se obtiene?

PREGUNTA 3 CONCURRENCIA (2.5 PUNTOS)

- a) ¿Qué implicaciones tiene el siguiente cambio en la implementación de un *lock* basado en swap atómico (x86)? ¿Sigue funcionando?

```
while(1){
    while(lock > 0);
    if(xchg(&lock, 1) == 0)
        return;
}
```

- b) ¿Por qué las tablas de hash concurrentes implementadas con listas enlazadas concurrentes exhiben una escalabilidad mejor que las listas enlazadas concurrentes?
- c) La siguiente función corresponde a la ejecución de un *thread*. El resultado de dicho *thread* no se retorna correctamente. Proponer una corrección al problema/s.

```
void *mythread(void *arg) {
    myarg_t *m = (myarg_t *) arg;
    myret_t r;
    r.x = 1;
    r.y = 2;
    return (void *) &r;
}
```

- d) ¿Cuál es la utilidad de las llamadas par k() y unpar k() en las colas de espera no activas?

PREGUNTA 4 PERSISTENCIA (2.5 PUNTOS)

- a) Describir los accesos a disco que requerimos para crear un fichero (que requiere un único bloque de datos) en el directorio /usr/share/doc.
- b) Suponiendo que el tamaño de bloque del sistema de ficheros es **4KB**, ¿Cuanto es el tamaño de fichero máximo que soportaría en VSFS, si cada i-nodo tiene **10** punteros directos y **1** puntero indirecto? Cada puntero tiene **32** bits (4 bytes).
- c) Si disponemos de un disco SSD de **128 GB**, con un *sparing* del 20%, y celdas NAND con un *endurance* limitado a **1000** ciclos de P/E. Los chips Flash están emplean páginas de **4 KB** y bloques de **512 KB** ¿Cuánto es el total de escrituras que aceptara el disco antes de que el *sparing* no sea suficiente para que el **FTL** contenga las celdas defectuosas (suponiendo que el FTL empleado logra un **wear-leveling perfecto**)? ¿Cómo cambiara dicha cantidad de escrituras si el **FTL** emplea mapeo entre páginas virtuales y páginas físicas a nivel de bloque?
- d) ¿Cuál es el cometido del CR en LFS? ¿Qué consecuencias puede tener una pérdida de potencia eléctrica en un sistema que emplea LFS sin *journaling*?

EXAMEN SISTEMAS OPERATIVOS AVANZADOS

LABORATORIO
FEBRERO 2018

NOMBRE Y FIRMA:

USUARIO GITLAB:

1. (2.5 punto) Sobre la base de la práctica 0, implementar la posibilidad de hacer la ordenación de las frases comenzando por el principio o por el final. Esto se deberá controlar con la opción “-r” (i.e. *fastsort -num -r fichero* indica que queremos ordenar el fichero por la posición núm contando desde el final de la línea, sin “-r” debería seguir ordenando por el principio).
2. (2.5 puntos) Sobre la base de la práctica 1, extender la llamada “**getprocs()**” a “**getprocs(int)**” de tal manera que podamos especificar en el parámetro el estado de los procesos en los que estamos interesados en contabilizar.
3. (2.5 puntos) Sobre la base de la práctica 2, extender la posibilidad de establecer 3 prioridades en los procesos (1 *low*, 2 *medium*, 3 *high*)
4. (2.5 puntos) Sobre la base de la segunda parte de la práctica 3 limitar la posibilidad de acceso a las páginas compartidas a procesos con PID igual o inferior al primer proceso que realice el acceso.
5. (2.5 puntos) Sobre la base de la práctica 4, implementar dentro de la librería de *threads* la función **cv_broadcast(cond_t *)**. No se permite implementar nuevas llamadas al sistema.
6. (2.5 punto) Sobre la base de la práctica 5, añadir que un fichero pequeño pase a ser normal cuando el tamaño supera la capacidad del *i-nodo*

Has de elegir **únicamente 4** de las preguntas propuestas. Es recomendable tener en cuenta el grado de complejidad de cada una de ellas en esta elección. Se dispone de **30 minutos** para cada una de ellas.

Cada solución debe estar en directorios independientes “*Lab/ExamenF/P[0-5]*” de tu rama de trabajo (DNI) en *gitlab* al finalizar la hora del examen. Cada respuesta tiene que estar en un *commit* independiente. Si vpuente@gmail.com no es “*Developer*” en tu repositorio, hazlo ya.