Examen parcial 1
Sistemas Operativos II

14/05/2015

Nombre, Apellido y Legajo: _____

1. Planificación en Linux. Indique verdadero o falso, justificando su respuesta.
   A. Todos los procesos listos tienen estado `TASK_RUNNING`.
   B. La prioridad estática se establece con `nice` o `set_priority`.
   C. El nuevo CFS del kernel 2.6.24 utiliza un árbol ordenado por `rq->fair_clock - vruntime`.

2. Modifique el siguiente ejemplo para que se utilice el manejador `handler_viejo` al recibir un `control-C`. No puede cambiar la llamada a `sigaction`:

```
#include <unistd.h>
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
/*
struct sigaction {
       void     (*sa_handler)(int);
       void     (*sa_sigaction)(int, siginfo_t *, void *);
       sigset_t   sa_mask;
       int        sa_flags;// SA_NOCLDSTOP SA_NOCLDWAIT SA_NODEFER
                           // SA_ONSTACK SA_RESETHAND SA_RESTART SA_SIGINFO
       void     (*sa_restorer)(void);
          };
*/
void handler_viejo(int sig) {
       printf("Bueno, estoy viejito pero todavia puedo hacer algo\n");
       exit(1);
}
static void hdl (int sig, siginfo_t *siginfo, void *context)
{
       printf("Bueno, ya que insisten voy a salir\n");
       exit(1);
}
```

```
int main (int argc, char *argv[])
{
        struct sigaction act;
        memset (&act, '\0', sizeof(act));
        act.sa_sigaction = &hdl;
        act.sa_flags = SA_SIGINFO;
        sigaction(SIGINT, &act, NULL);
        while (1)
                sleep (10);
        return 0;
}
```

3. Nombre las dos señales que no se pueden capturar. ¿En qué estado queda el proceso luego de recibir cada una?

4. Calcule el tiempo de respuesta para cada proceso si se utiliza:

   1. Multicolas con prioridad (con Round Robin dentro de cada cola de prioridad). Los procesos tienen prioridad fija (A=2, B=10, C=5, D=5) siendo 10 la mayor prioridad.
   2. Round Robin (sin prioridades)

   Para ambos casos suponga un quantum=2, que los tiempos de núcleo y del planificador son despreciables y que los procesos no se bloquean.

Proceso	Llegada	Duración
A	1	2
B	4	4
C	0	8
D	5	4

5. Responda las siguientes preguntas (justificando):

   1. Entre Round Robin y FIFO ¿ Qué algoritmo utilizaría para un sistema interactivo?
   2. Entre Shortest Process Next y Round Robin ¿ Qué algoritmo utilizaría para un sistema no interactivo?

Primer parcial de Sistemas Operativos II

1.A. Verdadero. El estado TASK_RUNNING en Linux corresponde a todos los procesos en ejecución y también a todos los procesos listos; no se hace distinción entre ambos estados a nivel de la identificación de estado de un proceso en Linux, aunque lógicamente existen en la práctica procesos que se están ejecutando en un momento dado y procesos que no.

1.B. Verdadero. La prioridad estática es la que puede establecer el usuario, y lo hace por medio de la llamada al sistema *setpriority*, el comando *nice* (para procesos a ejecutar) o el comando *renice* (para procesos ya ejecutándose). Como contraparte, la prioridad dinámica es establecida internamente por el núcleo y el usuario no la puede manipular directamente.

1.C. Falso. *rq->fair_clock* y *runtime* son 2 relojes/medidas de tiempo distintas y cada una de ellas se ha usado en una implementación distinta de CFS. *rq->fair_clock* se usaba en la implementación original y *runtime* se usa en la implementación nueva.

2. Se modifica solo la definición de la función *main*:

```
int main(void)
{

    struct sigaction act;
    memset(&act, 0, sizeof act);
    act.sa_handler = &handler_viejo;
    sigaction(SIGINT, &act, NULL);
```

```
        while (1)
            sleep(10);
        return 0;
    }
```

3. Las 2 señales de Unix que no se pueden capturar son SIGKILL y SIGSTOP. La primera termina el proceso inmediatamente, es decir que lo deja en estado "zombie". La segunda detiene el proceso de forma que luego pueda ser reanudado por medio del envío de SIGCONT; así que lo deja en estado bloqueado.

4.2. Gráfico correspondiente a RR:

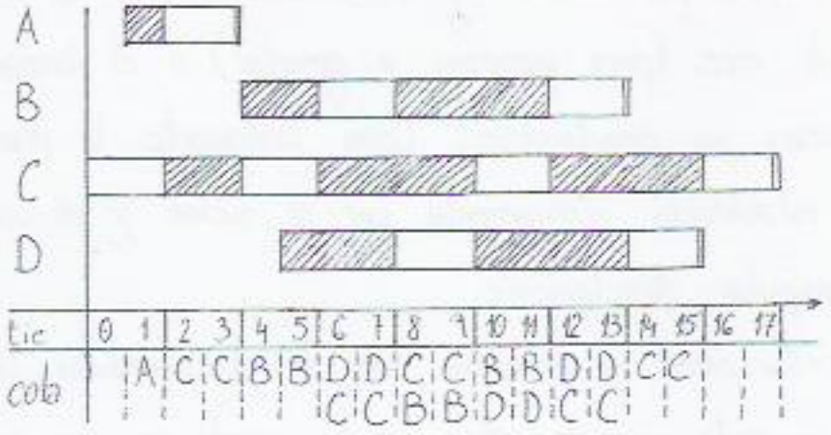


Tabla de tiempos correspondiente a RR:

Proceso	T
A	3
B	10
C	18
D	11

4.1. Gráfico correspondiente a multicolas con prioridad:

Parcial de Sistemas Operativos II

jue 14/05/2015

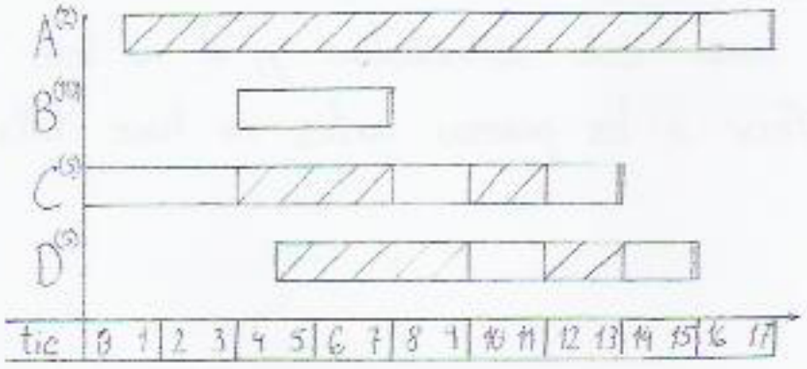


Tabla de tiempos correspondiente a multicolas con prioridad:

Proceso	T
A	17
B	4
C	14
D	11

5.1. Para un sistema interactivo es preferible el algoritmo RR en vez de FIFO. La diferencia entre ambos es que el segundo corresponde a un mecanismo de multitarea cooperativa, donde los cambios de contexto solo se dan cuando el proceso en ejecución se bloquea o cede el control, y el primer algoritmo es de multitarea apropiativa, la cual incorpora cambios de contexto forzosos cada cierto tiempo. Con FIFO, un proceso podría no soltar la CPU por largo tiempo mientras efectúa algún trabajo intensivo en uso de CPU y que no requiera entrada/salida, y los procesos interactivos tendrían que quedarse esperando a que este termine aun cuando ya estén disponibles los recursos que hayan solicitado. Esto significaría para el usuario retardos inaceptables, los procesos no respon-

derían fluidamente.

5.2. Para un sistema no interactivo, es preferible RR en vez de SPN. RR tiene menor costo administrativo y, al no tener que procurar beneficiar a los procesos cortos, no hace falta SPN.