Examen 1º parcial de Sistemas Operativos Avanzados (27/3/2019)

No se permite el uso de documentación. Duración: 90 minutos.

Nombre y apellidos:

Tema 1 (2,5 puntos) Suponga que un usuario normal ejecuta la siguiente secuencia de mandatos:

unshare -r -m -u -f --mount-proc -p bash

echo $$ # mandato 1

id # mandato 2

hostname MyBox # mandato 3

mount -t tmpfs tmpfs /mnt # mandato 4

cat /etc/shadow # mandato 5

Explique para cada uno de los 5 mandatos si se produce un error al ejecutarlo y, en caso contrario, qué labor realiza (qué valor imprime en caso de que esa sea su labor) y cómo esta viene condicionada por el uso de espacios de nombres identificando qué opción del mandato *unshare* afecta a cada mandato.

- **Mandato 1**: imprime el PID del proceso *bash* creado por el mandato *unshare*, que será igual a 1, al haberse especificado la opción *-p* en ese mandato, lo que causa que se cree un nuevo espacio de nombres de identificadores de proceso. El uso de la opción *-f* hace que el nuevo espacio de PID esté asociado al proceso hijo y no al proceso que ejecutó el mandato *unshare*.
- **Mandato 2**: imprime el UID y GID del usuario actual, que será 0 en ambos casos (lo que corresponde a un administrador del sistema), puesto que la opción *-r* ha creado un espacio de nombres de usuario donde el usuario normal que ejecutó el mandato *unshare* ejerce el papel de *super-usuario*. Gracias al uso de esta opción, un usuario normal tiene la capacidad de crear los restantes espacios de nombres que se especifican en ese mandato *unshare*.
- **Mandato 3**: cambia el nombre de la máquina, pero, debido al uso de la opción *-u*, esa modificación solo es visible dentro del nuevo espacio de nombres UTS creado por el mandato *unshare*
- **Mandato 4***:* monta un sistema de ficheros, pero, por el uso de la opción *-m,* ese montaje solo es visible dentro del nuevo espacio de nombres de montaje creado por el mandato *unshare.*
- **Mandato 5***:* Intenta acceder a un fichero accesible solo por el *super-usuario*, pero se produce un error ya que en el espacio de nombres por defecto donde reside ese fichero, el usuario que ejecutó el mandato *unshare* no tiene ningún privilegio especial.

Tema 4 (2,5 puntos) Con respecto al algoritmo CFS de Linux: **(a)** especifique cómo se calcula el tiempo de ejecución (la "rodaja") del proceso seleccionado por el planificador (*Pi* de peso *Wi*) suponiendo que el número de procesos listos (*N*, con un peso total de *W*) supera el umbral mínimo de granularidad; **(b)** suponiendo que el proceso elegido por el planificador ejecuta muy brevemente (unos pocos microsegundos) hasta bloquearse, produciéndose su desbloqueo de forma casi inmediata (nuevamente, unos pocos microsegundos), explique qué valor de *vruntime* se consideraría para ese proceso desbloqueado; **(c)** razone si es posible que un proceso en la cola de listos tenga un *vruntime* significativamente distinto (varios segundos mayor o menor) que el resto, para lo que debería tener en cuenta la procedencia de ese hipotético proceso listo (nuevo, desbloqueado o expulsado);

(a) Dado que el número de procesos es mayor que el umbral mínimo de granularidad, se ampliará el periodo de ejecución para asegurar esa granularidad mínima:

*periodo = sched_min_granularity * N*

con lo que el cálculo del tiempo de ejecución del proceso seleccionado por el planificador será:

*"Rodaja" = periodo * (Wi/W) = sched_min_granularity * N * (Wi/W)*

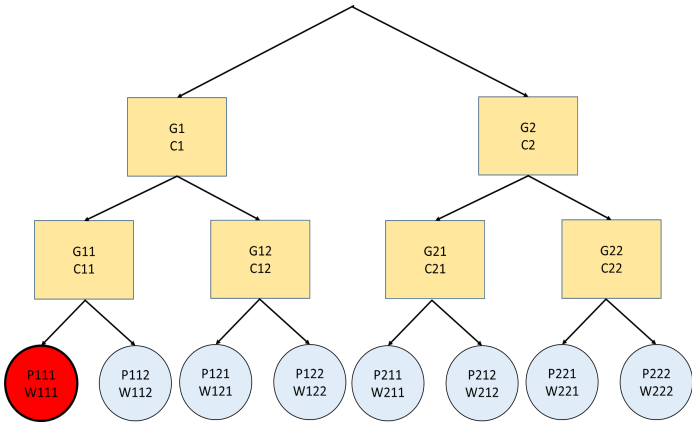
(b) Cuando se desbloquea un proceso se le asigna como *vruntime* el máximo entre el *vruntime* que tenía en el momento en el que se bloqueó y el mínimo *vruntime* actual en el sistema menos el valor de *sched_latency*, para, de esta forma, dar más prioridad a los procesos que se bloquean, pero asegurándose de que no acaparan el procesador:

max(vruntime, min_vruntime - sched_latency)

En el caso planteado en el enunciado, ha pasado muy poco tiempo desde que se le asignó el proceso al procesador (instante en el cual era el proceso listo en el sistema con menor *vruntime*) hasta que se desbloqueó de nuevo, después de un muy breve bloqueo. En consecuencia, como ha ejecutado muy poco tiempo, el *vruntime* del proceso cuando se bloqueó apenas habrá cambiado con respecto al momento en el que fue seleccionado y seguirá siendo el más bajo o uno de los más bajos entre los procesos listos, puesto que los *vruntime* del resto de procesos tampoco han podido casi cambiar ya que ha transcurrido muy poco tiempo. Como conclusión, se usará como nuevo *vruntime* para el proceso que se ha desbloqueado el valor que tenía cuando se bloqueó (primer término en la fórmula previa que calcula el máximo).

(c) Tal como opera el algoritmo CFS no puede haber discrepancias significativas entre los *vruntime* de los procesos listos para ejecutar. Por un lado, se intenta mantener el mismo *vruntime* para todos los procesos presentes en la cola de listos. En caso de que aparezca un nuevo proceso, se le asigna el mínimo *vruntime* en el sistema, con lo que se incorpora a la cola de listos con un *vruntime* similar al de sus compañeros. Por otro lado, si se desbloquea un proceso, el *vruntime* puede estar atrasado como máximo un tiempo *sched_latency*, que corresponde a unos pocos milisegundos, con respecto al del proceso que tenga el valor mínimo. Por último, al ser expulsado un proceso, su *vruntime* también es similar al de sus compañeros puesto que antes de asignarle el procesador era el que menos valor tenía.

(d) Suponiendo que se han definido dos grupos de procesos (*G1* de cuota/*share C1* y *G2* de cuota *C2*), tal que cada grupo tiene dos grupos hijos (por un lado, *G11* de cuota *C11* y *G12* de cuota *C12*, mientras que, por otro, *G21* de cuota *C21* y *G22* de cuota *C22*), cada uno de los cuales tiene a su vez dos procesos asociados (*P111* de peso *W111* y *P112* de peso *W112*; *P121* de peso *W121* y *P122* de peso *W122; P211* de peso *W211* y *P212* de peso *W212; P221* de peso *W221* y *P222* de peso *W222*), determine qué pesos y cuotas habría que tener en cuenta a la hora de calcular la "rodaja" de *P111*.



"Rodaja"= periodo **** W111/(W111+W112) * C11/(C11+C12) * C1/(C1+C2)***

Examen 1º parcial de Sistemas Operativos Avanzados (27/3/2019)

No se permite el uso de documentación. Duración: 90 minutos.

Nombre y apellidos:

Tema 2 y 3 (5 puntos)

(a) Dado un sistema monoprocesador con núcleo expulsivo, dibuje la traza de ejecución de dicho procesador para los procesos A, B y C en el siguiente escenario, indicando si los procesos están en modo usuario o modo sistema, si hay cambio de contexto voluntario o involuntario y las diferentes rutinas de eventos que tienen lugar, y teniendo en cuenta que:

- La rutina de interrupción de disco tiene asociada una rutina de interrupción software de sistema
- Sólo existen los procesos de usuario A, B, y C. La planificación de los procesos se hace por tiempo compartido (uso de rodaja de tiempo), siguiendo el orden A, B y C, en la medida de lo posible. Para simplificar el dibujo de la traza, considere que la rodaja de cada proceso es la misma y corresponde a 4 interrupciones de reloj y que dichas interrupciones están suficientemente espaciadas para no complicar demasiado la traza. Cada vez que un proceso se queda bloqueado, se "resetea" su rodaja de tiempo.
- La traza debería recoger al menos un cambio de contexto involuntario.
- El proceso A ejecuta una llamada *read*() sobre un fichero. Dentro de la rutina de dicha llamada se produce un fallo de página. El proceso A se quedará bloqueado. Eventualmente vendrá una interrupción de disco que despertará al proceso A, poniéndolo en ejecución de nuevo. El proceso A continuará en modo usuario y finalizará mediante la invocación a una llamada *exit*(). El proceso B ejecuta una llamada *wait*() sobre un semáforo que está cerrado, quedándose bloqueado. El proceso C es nuevo. Trabaja en modo usuario durante un tiempo y después realiza la llamada *post*(), abriendo el semáforo por el que estaba bloqueado B. El proceso C posteriormente continuará su ejecución en modo usuario. Cuando el proceso B se desbloquea y es planificado, ejecuta en modo usuario durante un gran intervalo de tiempo, realizando una división por cero, lo que provoca que el proceso finalice.

(b) Explique por qué un núcleo expulsivo está más preparado para tratar los problemas de sincronización existentes en un multiprocesador que un núcleo no expulsivo.