Universidad Nacional de Ingeniería
Facultad de Ciencias
Escuela Profesional de Ciencia de la Computación
Sistemas Operativos Avanzados (CC-571)
Ciclo: 2020-I

Segunda práctica calificada

Normas

- Presentar cada una de sus respuestas en un archivo separado. Los archivos separados tendran el nombre de solución1.c o solución2.mkd si es que es la respuesta es texto.
- En las respuestas de texto se tomará en cuenta la ortografía y la semántica de las soluciones.
- Todos los programas deben presentar pruebas de ejecución ya sea a través de imágenes o de video. Las respuestas sin ejecución serán calificados con 0.
- La entrega es en un archivo comprimido.
- No se admiten copias. **Cualquier evidencia de copia de otra fuente no serán consideradas y el puntaje será cero**.

Preguntas

1. Considera un sistema que implementa la planificación de colas multinivel. ¿Qué estrategia puede utilizar una computadora para maximizar la cantidad de tiempo de CPU asignado al proceso del usuario? Qué pasa si el sistema implementa round-robin?.
2. El tamaño de página en un sistema (que ejecuta un sistema operativo similar a Linux en hardware x86) se incrementa manteniendo todo lo demás (incluido el tamaño total de la memoria principal) igual. Para cada una de las siguientes métricas, indica si generalmente se espera que la métrica aumente, disminuya o no cambie como resultado de este aumento en el tamaño de la página:
   (a) Tamaño de la tabla de páginas de un proceso
   (b) Tasa de aciertos (hits) de TLB
   (c) Fragmentación interna de la memoria principal.
3. Discute cómo los siguientes pares de criterios de planificación entran en conflicto en ciertos entornos.
   (a) Uso de CPU y tiempo de respuesta
   (b) Tiempo de entrega promedio (TAT) y el tiempo de espera máximo
   (c) utilización de dispositivos de E/S y utilización de CPU.

4. Considera un proceso con 4 páginas lógicas, numeradas del 0 al 3. La tabla de páginas del proceso consta de las siguientes asignaciones de números de páginas lógicas a números de trama física: $(0, 11), (1, 35), (2, 3), (3, 1)$. El proceso se ejecuta en un sistema con direcciones virtuales de 16 bits y un tamaño de página de 256 bytes. Se garantiza que este proceso accede a la dirección virtual 770. Responde las siguientes preguntas, mostrando los cálculos adecuados.

   (a) ¿A qué número de página lógica corresponde esta dirección virtual?

   (b) ¿A qué dirección física se traduce esta dirección virtual?

5. Proporciona una ventaja de usar el asignador slab en Linux para asignar objetos del kernel, en lugar de simplemente asignarlos desde un heap de memoria dinámica.

6. Considera un sistema simple que ejecuta un solo proceso. El tamaño de los marcos físicos y las páginas lógicas es de 16 bytes. La RAM puede contener 3 marcos físicos. Las direcciones virtuales del proceso tienen un tamaño de 6 bits. El programa genera las siguientes 20 referencias de direcciones virtuales a medida que se ejecuta en la CPU: $0, 1, 20, 2, 20, 21, 32, 31, 0, 60, 0, 0, 16, 1, 17, 18, 32, 31, 0, 61$. (Nota: las direcciones de 6 bits se muestran en decimal aquí.) Suponga que los marcos físicos en la RAM están inicialmente vacías y no se asignan a ninguna página lógica.

   (a) Traduzca las direcciones virtuales anteriores a números de página lógicos a los que hace referencia el proceso. Es decir, anote la cadena de referencia de 20 números de página correspondientes a los accesos de direcciones virtuales anteriores. Supongamos que las páginas están numeradas a partir de $0, 1, \ldots$

   (b) Calcula el número de fallas de página generadas por los accesos anteriores, suponiendo un algoritmo de reemplazo de página FIFO. También debes indicar correctamente qué accesos de página en la cadena de referencia que muestra en la parte (a) son responsables de las fallas de la página.

   (c) Repite (b) para el algoritmo de reemplazo de página LRU.

   (d) ¿Cuál sería el menor número de fallas de página alcanzables en este ejemplo, suponiendo que se utilizara un algoritmo de reemplazo de página óptimo? Repite (b) para este algoritmo óptimo.

7. Considera un asignador de memoria que utiliza el algoritmo de asignación buddy para satisfacer las solicitudes de memoria. El asignador comienza con un heap de tamaño 4KB (4096 bytes). El programa de usuario realiza las siguientes solicitudes al asignador (todos los tamaños solicitados están en bytes): `ptr1 = malloc (500); ptr2 = malloc (200); ptr3 = malloc (800); ptr4 = malloc (1500)`. Si el encabezado agregado por el asignador tiene menos de 10 bytes de tamaño. Puedes hacer cualquier suposición sobre la implementación del algoritmo de asignación buddy que sea consistente con la descripción en clase.

   (a) Dibuja una figura que muestre el estado del heap después de completar estas 4 asignaciones. Tu figura debe mostrar qué partes del heap están asignadas y cuáles son libres, incluidos los tamaños de los diversos bloques asignados y libres.

   (b) Ahora, supongamos que el programa de usuario libera asignaciones de memoria de `ptr2`, `ptr3` y `ptr4`. Dibuja una figura que muestre el estado del heap una vez más, después de que se libere memoria y el algoritmo de asignación haya tenido la oportunidad de realizar cualquier posible coalescing.

8. Considera un sistema con direcciones virtuales y físicas de 8 bits y páginas de 16 bytes. Un proceso en este sistema tiene 4 páginas lógicas, que se asignan a 3 páginas físicas de

la siguiente manera: la página lógica 0 se asigna a la página física 6, 1 se asigna a 3, 2 se asigna a 11 y la página lógica 5 no se asigna a ninguna página física todavía. Todas las otras páginas en el espacio de direcciones virtuales del proceso están marcadas como inválidas en la tabla de páginas. La MMU recibe un puntero a esta tabla de páginas para la traducción de direcciones. Además, la MMU tiene una pequeña caché TLB que almacena dos entradas, para las páginas lógicas 0 y 2. Para cada dirección virtual que se muestra a continuación, describe qué sucede cuando la CPU accede a esa dirección. Específicamente, debes responder lo que sucede en el TLB (¿hit o miss?), MMU (¿a qué entrada de la tabla de página se accede?), SO (¿hay algún tipo de trap?) y la memoria física (¿a qué dirección física se accede?). Escribe la dirección física traducida en formato binario. (Ten en cuenta que no está implícito que los accesos a continuación sucedan uno tras otro; debes resolver cada parte de la pregunta de forma independiente utilizando la información proporcionada anteriormente).

(a) Dirección virtual 7

(b) Dirección virtual 20

(c) Dirección virtual 70

(d) Dirección virtual 80

9. Resolver

   (a) En este ejercicio estudiamos la afinidad de caché. Para hacer esto, necesitarás usar el indicador $-A$. Este indicador se puede utilizar para limitar en qué CPU el planificador puede colocar un trabajo en particular. En este caso, lo debes usar para colocar los trabajos `b` y `c` en la CPU 1, mientras restringimos `a` a la CPU 0. Esto se logra al escribir como:

   ```
   python multi.py -n 2 -L a: 100 : 100, b: 100: 50, c: 100: 50 -A
                   a: 0, b: 1, c: 1;
   ```

   ¿Puedes predecir qué tan rápido se ejecutará esta versión?. ¿Por qué lo hace mejor?. ¿Qué otras combinaciones de `a`, `b` y `c` en los dos procesadores funcionarán más rápido o más lento?. No olvides que en tus respuestas debes activar varias opciones de seguimiento para ver lo que realmente está sucediendo y como se relaciona con la afinidad de caché.

   (b) Un aspecto interesante de los multiprocesadores de almacenamiento en caché es la oportunidad de acelerar los trabajos mejor de lo esperado cuando se usan con múltiples CPU (y sus cachés) en comparación con la ejecución de trabajos en un solo procesador. Específicamente, cuando se ejecuta en N CPU, a veces puede acelerar más de un factor de N, una situación denominada aceleración superlineal.

   Para experimentar con esto, usa la descripción del trabajo aquí (`-L a: 100: 100, b: 100: 100, c: 100: 100`) con un pequeño caché (`-M 50`) para crear tres trabajos. Ejecuta esto en sistemas con 1, 2 y 3 CPU (`-n 1, -n 2, -n 3`). Ahora, haz lo mismo, pero con un caché por CPU más grande de tamaño 100. ¿Qué observas sobre el rendimiento a medida que aumenta el número de CPU? Usa -c para confirmar sus conjeturas y otros indicadores para presentar mejor tu respuesta.

10. Considera un proceso con 9 páginas lógicas, de las cuales 3 páginas se asignan a marcos físicos. El proceso accede a una de sus 9 páginas al azar. ¿Cuál es la probabilidad de que el acceso resulte en un hit de TLB y un error de página posterior?.