
Laboratorio

Procesos en un entorno de Linux

Fecha de asignación: 27,28 de Agosto, 2024
Grupos: 2 personas

Fecha de entrega: 3,4 de Setiembre, 2024
Profesor: Jason Leitón, Leonardo Araya

1. Objetivos

- Identificar algunas funciones con las cuales se puede interactuar con los procesos en un entorno de Linux con el fin de diagnosticar los problemas más comunes.
- Comprender el significado de los diferentes parámetros de los procesos.
- Evaluar diferentes valores para los procesos en un entorno de Linux.

2. Indicaciones

1. Para el siguiente laboratorio deberá ejecutar los pasos que se detallan en cada uno de los apartados, comprobando su ejecución con capturas de pantalla en el resultado del mismo. Esta información deberá estar en un documento PDF para cargarlo en Tecdigital según corresponda.
2. Se debe de realizar la guía de preguntas y adjuntar las soluciones junto con enunciados en el mismo documento del punto anterior.
3. La fecha de entrega será la indicada en este documento y debe ser de manera individual.
4. En caso de que el laboratorio necesite código fuente, este también debe de incorporarse como parte de la solución del mismo, ya que será evaluado.
5. El laboratorio debe ser revisado por el profesor antes de la fecha de cargar los archivos, por lo que el estudiante será el encargado de mostrar su trabajo, en caso de que no lo haga la nota será cero.

3. Preguntas guía

1. Explique las etapas de creación de un proceso en Windows (CreateProcess).
2. ¿Cuáles son las variables necesarias que se deben de guardar cuando se quiere implementar un cambio de contexto?

3. ¿Cómo se podría implementar un cambio de contexto por hardware y no por software? Realice un esquema de arquitectura con su propuesta.
4. Para que sirve el comando ps y top en un entorno de Linux.
5. Investigue los posibles estados de un proceso en un entorno de Linux y cómo se representan.

4. Procesos en Linux

1. Conéctese a su máquina virtual por medio de SSH (**También lo puede hacer local**). Acceda mediante dos conexiones, es decir, dos consolas.
2. Ejecute el comando: ps -aux (obtenga la captura de los últimos ítem) explique cual es el significado de aux.
3. Investigue cada uno de los datos de los procesos del punto anterior (PID, VSZ ...).
4. Busque el comando que retorna los procesos propios de un usuario y tome la captura de pantalla del que posee mayor tiempo en el procesador.
5. Ejecute el comando: top en la primera consola. ¿Para que sirve?
6. Ejecute el comando (5 veces en la segunda consola): cat /dev/zero > /dev/null &
7. Ejecute el comando top nuevamente en la primera consola. ¿Qué observa con respecto al top anterior?
8. ¿Qué significa los valores de cada uno de los parámetros de los procesos creados (PR, NI, VIRT ...)?
9. Note que todos los procesos creados tienen una prioridad similar ¿Por qué sucede esto?
10. ¿ Por qué el parámetro “Time ” aumenta paulatinamente ?
11. Obtenga un identificador de alguno de los procesos creados anteriormente.
12. Aumente la prioridad de dicho proceso con el comando: renice -n 10 PID.
13. Inicie un proceso con prioridad alta con el siguiente comando: nice -n -10 cat /dev/zero > /dev/null &
14. Ejecute el comando top nuevamente.

5. Creación de procesos con parámetros establecidos por el usuario

Para los incisos de gráficos pueden utilizar excel o alguna herramienta similar, la idea es solo introducir los datos arrojados por la ejecución y generar automáticamente el gráfico.

1. Para los siguientes programas pruebe su funcionamiento con valores que tomen algún tiempo considerable en terminar su ejecución, por ejemplo, valores mayores a 100.
2. Realice un programa recursivo en C y Python que sea capaz calcular el factorial de cualquier número entero.
3. Ejecute dichos programas en su máquina virtual y tome el tiempo de ejecución, así como los parámetros de ambos procesos con el comando top (En caso de que sea necesario coloque prints entre cada recursión).
4. Ejecute el programa realizado en Python 5 veces con 5 diferentes prioridades de manera ascendente y muestre una gráfica del comportamiento del mismo (Prioridad-Tiempo).
5. Ejecute el programa realizado en C 5 veces con 5 diferentes prioridades de manera descendente y muestre una gráfica del comportamiento de mismo (Prioridad-Tiempo).
6. Investigue el comando para eliminar un proceso. Posteriormente ejecute el programa en C y elimínelo antes de que termine su ejecución.
7. Discuta el comportamiento de la gráfica. ¿Es el comportamiento que esperaba?

En caso de que su máquina virtual o local realice el procesamiento de manera rápida, puede utilizar un print en cada recursión para aumentar el tiempo de ejecución.

6. Hilos en Linux

Para las pruebas de esta sección utilice archivos con gran contenido de texto (por ejemplo, El quijote, la odisea), la idea es exigir la máquina.

1. Realice un programa en C que tome un archivo de texto (.txt) y cuente la cantidad de apariciones de una determinada palabra. Tome el tiempo de ejecución del mismo.
2. Implemente dicho programa con 2,3,4,5 hilos y grafique el comportamiento (Cantidad de hilos- Tiempo) puede utilizar un software como excel o su equivalente.
3. ¿La mejora es lineal? Justifique dicha monotonía.