



Práctica 1

Control, Creación y Monitorización de Procesos

Objetivos

- Poner en práctica los conocimientos sobre el Kernel de Linux.
- Familiarizarse con la terminal de Linux y comandos de sistema y usuario.
- Aprender a crear, monitorizar y montar procesos del Kernel de Linux.

Contexto

Más del 50% de los servidores utilizados para construir los servicios de internet utilizan el sistema operativo Linux. Debido a que es un sistema seguro y de código abierto, las empresas lo utilizan por sus garantías y bajo costo.

Los servicios contruidos sobre Linux abarcan un amplio espectro de soluciones; sin embargo, sin importar los por menores de los procesos levantados en el servidor nos vemos en la necesidad de implementar una forma de monitorear los recursos del sistema.

Solución Web de Monitorizaicón de Procesos y Memoria

Se deberá implementar un servidor web en Golang, creando una API simple para obtener información sobre la memoria y CPU del sistema haciendo uso de módulos del Kernel escritos en C para obtener la información y escribirla en archivos dentro de la carpeta /proc para posteriormente leerlos con Golang.

Queda a discreción del estudiante cuál tecnología utilizar para la interfaz gráfica que deberá ser visible a través del navegador, puede ser por medio de una IP y renderizando el html desde el servidor, o bien por medio de un sitio web estático alojado por ejemplo en el servicio S3 de Amazon. Los datos deben actualizarse cada 3 segundos automáticamente.

Monitor de Memoria

Se deberá crear y montar un módulo del Kernel llamado `mem_grupo<no_grupo>` con el cuál se debe extraer información sobre el uso de memoria en el sistema. Al montarse el módulo se debe imprimir el mensaje “Hola mundo, somos el grupo `<no_grupo>` y este es el monitor de memoria”. Al desmontarse el módulo se debe imprimir el mensaje “Sayonara mundo, somos el grupo `<no_grupo>` y este fue el monitor de memoria”

El monitor de memoria debe mostrar la información del consumo de RAM del servidor, en el cual se podrá visualizar la siguiente información:

- Total de memoria RAM del servidor (en MB)
- Total de memoria RAM consumida (en MB)
- Porcentaje de consumo de memoria RAM
- Gráfica en tiempo real del consumo de memoria RAM

Se debe crear una gráfica que debe mostrarse como un polígono de frecuencias en tiempo real del consumo de memoria del servidor, la librería a utilizar para crear esta gráfica queda a discreción del estudiante.

Administrador de Procesos y Arbol de procesos

Éste mostrará de manera tabulada todos los procesos que están siendo ejecutados en el servidor, así como un resumen de los procesos. La información general que se debe mostrar es la siguiente:

- Total de procesos: cantidad total de procesos registrados
- Procesos en ejecución: cantidad de procesos en estado *running*
- Procesos suspendidos: cantidad de procesos en estado *sleeping*
- Procesos detenidos: cantidad de procesos en estado *stopped*
- Procesos zombies: cantidad de procesos en estado *zombie*

Además debe mostrar de manera tabulada la siguiente información de cada proceso:

- PID: identificador del proceso
- Nombre del proceso
- Estado: estado en el que se encuentra el proceso
- %RAM: porcentaje de utilización de RAM del proceso
- Task Codesize: memoria ocupada por el código del proceso
- Usuario: nombre de usuario que ejecutó el proceso

Además de la información anterior, esta página también debe de contar con la opción de dar *KILL* a un proceso para terminar con el mismo, se verificará el correcto funcionamiento tanto en la actualización de la lista de procesos, como en la consola del servidor.

Se deberá mostrar un árbol de procesos dinámico, es decir, se debe mostrar una lista de procesos que permita seleccionar uno y desplegar a todos sus hijos con PID y nombre.

Para poder visualizar de una mejor manera el funcionamiento de todo lo anterior, utilizaremos el programa Stress, el cual es una herramienta de Linux la cual nos permite, mediante líneas de comando hacer pruebas al servidor, por ejemplo:

```
# stress -c 2 -i 1 -m 1 -- vm-bytes 128M -t 10s
```

Consideraciones

- La interfaz de usuario de la solución web debe ser amigable para que su utilización sea cómoda y fácil de utilizar.
- La distribución de Linux a utilizar queda a discreción del estudiante.
- La obtención de la información debe hacerse obligatoriamente por medio de los módulos de kernel y las respectivas estructuras que la contienen. Prohibido el uso de archivos ya existentes en el directorio /proc.
- Cualquier copia total o parcial será reportada a la Escuela de Sistemas para que proceda como corresponde.
- La práctica es en tríos utilizando los establecidos en el laboratorio.

Entregables

- Código fuente de la solución web.
- Código fuente de los módulos desarrollados.
- Manual de usuario y manual técnico.

El manual técnico debe contener la descripción del código de los módulos de la manera más detallada posible ya que esta será la prueba de que han entendido el funcionamiento de los módulos de kernel.

Forma de Entrega

Todo los entregables comprimidos en un archivo .zip con el nombre:

[SO2]Practica1_<no_grupo>. Todo debe ir bien identificado con los carnets de los 3 integrantes. Es indispensable que se realice la entrega de 2 formas:

- Mediante UEDI subiendo el archivo zip

- Subiendo todo a un repositorio de GitHub el cuál debe ser privado con el nombre: practica1_grupo<no_grupo>_so2. Se deberá agregar al usuario: *brayan-chinchilla*.

Se podrá calificar bien sea utilizando el archivo enviado a UEDI o clonando el repositorio de GitHub, se verificarán las fechas de última actualización de los archivos.

Fecha de Entrega: Miercoles 8 de Septiembre hasta las 23:59.