

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Sistemas Operativos 2
Escuela de Vacaciones junio 2023
Ing. Edgar Rene Ornelis Hoil
Aux. Bernald Renato Paxtor Perén
Sección A



Práctica 1

Control, Creación y Monitoreo 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.

Descripción

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.

Para esta práctica es necesario crear una máquina virtual de Linux sin interfaz gráfica, solamente con la opción de interactuar por medio de la consola, esto debido a que al trabajar con servidores en la nube para crear servicios web, la gran mayoría de estos servidores no tienen instalada una interfaz gráfica para ahorrar recursos del sistema y dedicar la mayor cantidad del poder computacional para ejecutar el servicio que es funcional para la empresa. Es por ello que una forma común de interactuar con estos servidores es por medio de una terminal remota utilizando el puerto SSH.

El sistema operativo debe de poder conectarse a internet. El nombre de la máquina debe de ser: grupo<no_grupo>-ing-usac-so2.

Crear una carpeta en el directorio raíz (/) llamada practica1, de la cual todos los integrantes del grupo puedan tener permisos de lectura, escritura y ejecución.

Además de la instalación, se requiere que las siguientes aplicaciones/paquetes estén instaladas en la distribución:

- GCC

- Make
- Linux-headers (La versión del kernel de la distribución)
- Java
- Golang
- Git

Solución Web del Monitor 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 Árbol 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
- Usuario: el usuario que ejecutó el proceso
- Estado: estado en el que se encuentra el proceso
- %RAM: porcentaje de utilización de RAM del 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, pero no debe tener interfaz gráfica, usualmente las versiones “server” cumplen con este requisito.
- 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
- Código fuente de los módulos desarrollados
- Manual de usuario y manual técnico de la solución

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]Practical_<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/rar
- Por medio de un repositorio de GitHub, el cual debe ser privado con el nombre: so2_practical_<<no_grupo>>.

Fecha de Entrega: Jueves 8 de Junio hasta las 23:59