



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y TECNOLOGÍA AVANZADA
CURSO DE ADMINISTRACIÓN DE SISTEMAS OPERATIVOS
GRUPO 1TM1 AGOSTO-DICIEMBRE 2016.
IMPARTIDO POR DR. MOISÉS SALINAS ROSALES

PRÁCTICA 2 DE LABORATORIO: GESTIÓN DE RECURSOS DE MEMORIA Y PROCESADOR EN LINUX CENTOS

1. INTRODUCCIÓN.

Esta práctica abordará el tema de gestión de recursos de memoria y procesador dentro de un sistema Linux, específicamente desde la perspectiva del control de procesos en ejecución en el sistema y de los recursos en uso por estos.

Este documento presenta una guía para el desarrollo de una práctica de laboratorio del Curso de Administración de Sistemas Operativos. Las secciones 2,3 y 4 describen un marco de referencia el ejercicio a realizar, así como los objetivos y requisitos para la misma. La secciones 5 y 6 son deberan ser desarrolladas por el estudiante conforme a lo que se indica en cada sección.

Para efectos de evaluación de esta práctica se deberá enviar el reporte de la misma, el cual consistirá en este documento guía anexando el resultado del desarrollo de lo solicitado en las secciones 5 y 6.

2. MARCO DE REFERENCIA.

Cuando se ejecuta un programa dentro de un sistema de cómputo el sistema operativo crea una instancia del mismo llamada proceso, a la cual se le asigna un espacio en memoria RAM donde se alojan tres elementos distintivos: una copia del *bytecode* dicho programa, un espacio para almacenamiento de datos que dicho programa utilice (como las variables u objetos creados dentro del programa), y un bloque donde el operativo almacena información para el control de proceso, llamado *Bloque de Control de Proceso* o PCB por sus siglas en inglés. Al conjunto de estos elementos en memoria se le conoce como *imagen del proceso* y una representación gráfica del mismo se muestra en la figura 1.



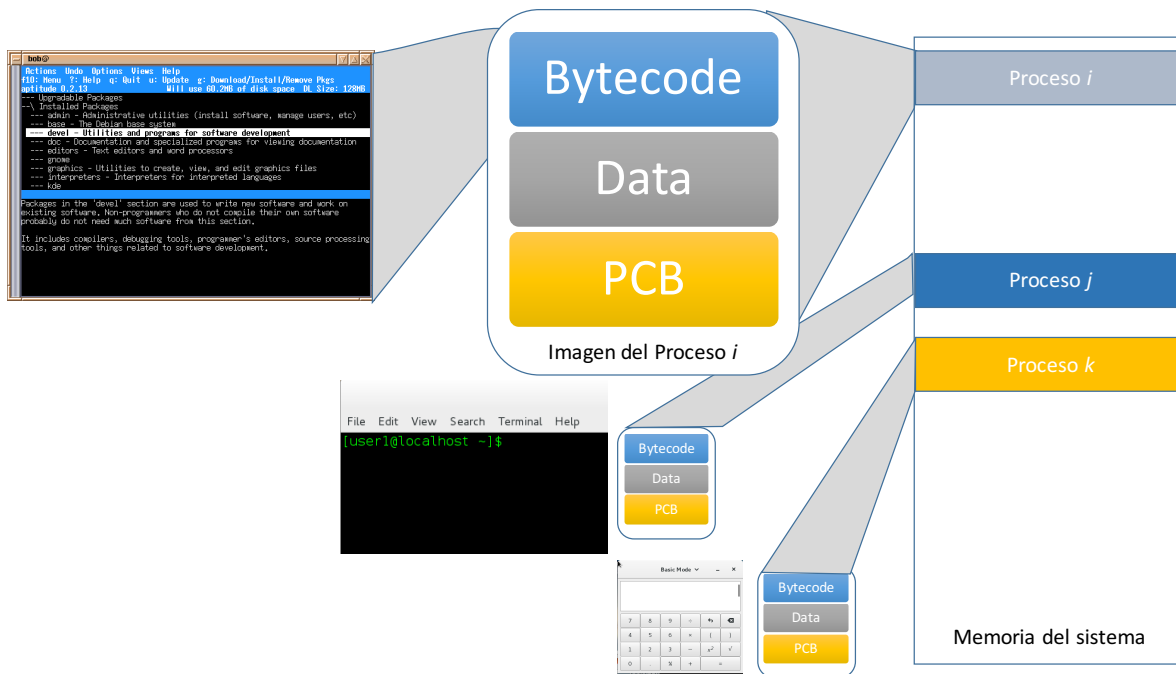


Figura 1. Imagen de proceso.

Bajo el concepto de proceso el sistema operativo asigna los recursos del hardware y software a las aplicaciones que se ejecutan en el equipo. La ejecución de dichas aplicaciones se realiza mediante la invocación al código ejecutable o *bytecode* que contiene las instrucciones que el microprocesador deberá ejecutar a fin de realizar las tareas asociadas a la aplicación.

Los recursos a los que el sistema operativo puede otorgar acceso a un proceso se clasifican de tres tipos: acceso al procesador, espacio en memoria y entrada/salida, tal como se describe a continuación:

Acceso al procesador: Cada proceso requiere de acceder a un procesador para ejecutar su código. Este acceso lo administra el sistema operativo bajo la modalidad de préstamo de tiempo ejecución en un procesador, para lo cual el sistema operativo establece una *política de planificación*; esta política define la forma en que se asigna a cada proceso permiso para utilizar un procesador por un tiempo determinado. A dicho tiempo se denomina *quantum* y tiene una duración que el diseñador del operativo define; en el caso de Linux, este puede variar según la prioridad del proceso desde 5 hasta 800 milisegundos (ms.), siendo 100 ms. el valor estándar.

Según del sistema operativo que se trate, existen varios tipos de políticas de planificación, siendo una de las más comunes la denominada *Ronda de Robin (RR)*; la cual retoma la



filosofía de compartir los recursos de manera equitativa entre todos los participantes, por lo que a cada proceso se le asigna un periodo de tiempo de préstamo de igual duración que el de los demás. Como ya se mencionó a dicho periodo se le conoce como *quantum*. Así pues, cada vez que un proceso demanda acceso al procesador este debe formarse en una cola en espera de ser atendido, una vez que *gana* acceso al procesador ejecutará tantas instrucciones como se a posible mientras dure el *quantum*, si al agotar este periodo aún quedan instrucciones pendientes por ejecutar, el proceso deberá formarse de nuevo en la cola correspondiente. Otras políticas comunes son la *First Come First Served (FCFS)* y las variantes que incorporan prioridades. Para mayor referencia se puede consultar [3].

Linux implementa una política de planificación de *Ronda de Robin con prioridades*, la cual incorpora varias colas donde los procesos son formados según la prioridad que se les asigne por el sistema. La prioridad de un proceso común corresponde a un valor 20, la cual puede ser incrementada por el sistema hasta llegar a un valor de 30, o bien puede ser decrementada hasta llegar a un valor de 0.

Dentro de la gestión de recursos de procesamiento que el administrador del sistema operativo Linux puede realizar para cada proceso destacan el cambio de estado: la terminación suspensión y recuperación de este, el ajuste de prioridad y el establecimiento de cuotas de tiempo máximo en uso del procesador.

Espacio en memoria: Cada proceso requiere que el sistema operativo le asigne un espacio en memoria RAM, en éste se alojará la imagen del proceso la cual se compone de tres secciones: código ejecutable o *bytecode*, datos del programa y el bloque de control del proceso. Si bien el código ejecutable se carga en un inicio, la sección con datos del programa se va asignando poco a poco durante la ejecución del programa conforme el mismo lo va solicitando al sistema operativo, en esta sección se incluye el espacio asignado de manera estática (variables y heap) y dinámica (llamadas a malloc).

Cada vez que Linux asigna espacio de memoria a un proceso, dicho espacio hace referencia a un bloque de direcciones dentro de la memoria RAM que se ponen a disposición para su uso por parte del proceso. Este bloque de direcciones está organizado Para mayor referencia se puede consultar [4] y [5].

Dentro de la gestión de memoria que el administrador del sistema operativo Linux puede realizar para cada proceso destacan la terminación o suspensión de procesos *abusivos*, así como el establecimiento de cuotas de tiempo máximo en uso del procesador.





Entrada/Salida: Los recursos de entrada y salida con todos aquellos que permiten el ingreso, exportación, visualización, transmisión y recepción de información de los procesos hacia otros procesos (locales o remotos) o bien hacia dispositivos que interactúen con el usuario, como las terminales.

3. OBJETIVO.

El objetivo de esta práctica es realizar un conjunto de ejercicios en torno a la gestión de procesos, para controlar su ejecución, ajustar sus prioridades, así como limitar el uso de recursos del mismo.

4. REQUERIMIENTOS DE LA PRACTICA

- Estación de trabajo con sistema operativo Linux Centos 7 y ambiente gráfico en operación.
- Acceso a la cuenta de administración root.
- Conexión a internet.

5. DESARROLLO

Al estudiante: Deberá complementar esta sección conforme al desarrollo de las siguientes actividades durante el ejercicio de la práctica en el laboratorio, documentando siempre los pasos, resultados y experiencias resultantes.

Las actividades a desarrollar como parte de esta práctica son las siguientes:

- 5.1. Consultar de los procesos que se están ejecutando en el sistema (instantánea).
- 5.2. Identificar la información asociada al uso de recursos de un proceso.
- 5.3. Consultar de la genealogía de un proceso.
- 5.4. Terminar de forma arbitraria un proceso.
- 5.5. Modificar el estado de un proceso mediante el envío de señales.
- 5.6. Ajustar la prioridad de un proceso.
- 5.7. Desvincular procesos anidados.
- 5.8. Establecer cuotas de uso de recursos para un proceso.

6. CONCLUSIONES

Al estudiante: Deberá completar esta sección a manera de conclusiones de haber realizado lo descrito en la sección 4. No olvide que las conclusiones son individuales y tienen la finalidad de capturar su impresión entorno a las experiencias y aprendizaje obtenidos durante el desarrollo de la práctica.





7. REFERENCIAS

- [1] Steven M. Bellovin, Slides on Operating Systems I, Disponible desde: <https://www.cs.columbia.edu/~smb/classes/s06-4118/l13.pdf>.
- [2] Yogesh Babar, Understanding Linux Process States, RedHat, Disponible desde: https://access.redhat.com/sites/default/files/attachments/processtates_20120831.pdf.
- [3] Paul Krzyzanowski, Operating systems course, Process Scheduling Who gets to run next?, Disponible desde: <https://www.cs.rutgers.edu/~pxk/416/notes/07-scheduling.html>.
- [4] Paul Krzyzanowski, Operating systems course, Memory Management, Disponible desde: <https://www.cs.rutgers.edu/~pxk/416/notes/09-memory.html>.
- [5] Paul Krzyzanowski, Operating systems course, Memory Management – Part 3 Kernel Memory Allocation, Disponible desde: <https://www.cs.rutgers.edu/~pxk/416/notes/content/11-kmem-slides.pdf>.

8. CONSULTAR RECOMENDADA DE HERRAMIENTAS DE LINUX

Como parte del desarrollo de la práctica deberá realizar algunas acciones como copiar, mover, renombrar archivos, así como visualizar y editar el contenido de archivos, así como otras acciones comunes en la administración de Linux. Para facilitar su tarea se recomienda revisar los manuales de ayuda del sistema para las siguientes herramientas:

ps top pstree kill grep more

Recuerde que para consultar los manuales del sistema se hace uso de la herramienta **man**, por ejemplo la siguiente instrucción despliega la ayuda del program cp:

```
$ man ps
```

