



**UTMACH**

UNIDAD ACADÉMICA DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TEMA:

IMPLEMENTACIÓN DE UNA NUEVA FUNCIONALIDAD AL KERNEL DE LA  
DISTRIBUCIÓN DE LINUX CENTOS MEDIANTE LLAMADAS AL SISTEMA

TRABAJO PRÁCTICO DEL EXAMEN COMPLEXIVO PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERA DE SISTEMAS

AUTORA:

PARRA LAMBERT ANGIE ESTEFANY

MACHALA - EL ORO

## CESIÓN DE DERECHOS DE AUTOR

Yo, PARRA LAMBERT ANGIE ESTEFANY, con C.I. 0705858587, estudiante de la carrera de INGENIERÍA DE SISTEMAS de la UNIDAD ACADÉMICA DE INGENIERÍA CIVIL de la UNIVERSIDAD TÉCNICA DE MACHALA, en calidad de Autora del siguiente trabajo de titulación IMPLEMENTACIÓN DE UNA NUEVA FUNCIONALIDAD AL KERNEL DE LA DISTRIBUCIÓN DE LINUX CENTOS MEDIANTE LLAMADAS AL SISTEMA

- Declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional. En consecuencia, asumo la responsabilidad de la originalidad del mismo y el cuidado al remitirme a las fuentes bibliográficas respectivas para fundamentar el contenido expuesto, asumiendo la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera EXCLUSIVA.
- Cedo a la UNIVERSIDAD TÉCNICA DE MACHALA de forma NO EXCLUSIVA con referencia a la obra en formato digital los derechos de:
  - a. Incorporar la mencionada obra al repositorio digital institucional para su democratización a nivel mundial, respetando lo establecido por la Licencia Creative Commons Atribución-NoComercial-Compartir Igual 4.0 Internacional (CC BY-NC-SA 4.0), la Ley de Propiedad Intelectual del Estado Ecuatoriano y el Reglamento Institucional.
  - b. Adecuarla a cualquier formato o tecnología de uso en internet, así como incorporar cualquier sistema de seguridad para documentos electrónicos, correspondiéndome como Autor(a) la responsabilidad de velar por dichas adaptaciones con la finalidad de que no se desnaturalice el contenido o sentido de la misma.

Machala, 19 de noviembre de 2015



---

**PARRA LAMBERT ANGIE ESTEFANY**  
**C.I. 0705858587**

# IMPLEMENTACIÓN DE UNA NUEVA FUNCIONALIDAD AL KERNEL DE LA DISTRIBUCIÓN DE LINUX CENTOS MEDIANTE LLAMADAS AL SISTEMA.

Angie Estefany Parra Lambert  
C.I.: 0705858587

## RESUMEN

Los sistemas operativos cada día se van adaptando a las necesidades de los usuarios, lanzando versiones para cada gusto y que permitan la mejor solución de los problemas, como también la mejor calidad al momento del uso de los recursos del sistema. ¿Pero, qué pasaría si los programas de usuario tienen necesidades específicas que no están implementadas en nuestros sistemas operativos? Las distribuciones de Linux, al ser software libre, nos permite mejorar nuestros sistemas y adicionarle nuevas funciones, éstos nuevos procedimientos que se implementarán han tomado el nombre de “llamadas al sistema”. Una llamada al sistema es un mecanismo cuyo nombre y número es único, el cual, se encarga de invocar un proceso para solicitar un servicio al sistema operativo, es decir, el usuario solicita una tarea específica y nueva por medio de un programa de usuario y este se ejecutará realizando la llamada al sistema que asignara los recursos disponibles y necesarios. La implementación de llamadas al sistema en un nuevo kernel de Linux mediante una compilación del mismo es un proceso muy fácil, pero sin embargo no debe ser minimizado como una tarea simple, porque, se debe tener en consideración conocimientos en lenguaje de programación C, manejo de librerías, manipulación de archivos de configuración de procesos, archivos que apoyaran el proceso de arranque, lo que se pretende es que, al momento de iniciar nuestro sistema operativo trabajemos en un entorno personalizado con nuevas funcionalidades. Las empresas dedicadas a solucionar problemas informáticos, podrán trabajar no solo en el área de desarrollo de software sino a nivel de sistemas operativos personificados para los usuarios, permitiendo trabajar en entornos nuevos con procedimientos de áreas específicas y mejor uso de recursos.

**Palabras clave:** Llamada al sistema, implementación, kernel, software libre, Linux.

# IMPLEMENTATION OF A NEW FUNTIONALITY TO THE KERNEL IN THE LINUX CENTOS DISTRIBUTION VIA SYSTEM CALLS.

Angie Estefany Parra Lambert  
I.D.: 0705858587

## ABSTRACT

Every day, operating systems are adapted to user needs, releasing new versions for each tilt and allow the best solution of problems, but, what if user has specific needs that aren't implemented in our operating system? Linux distributions are free software, allow to improve our systems and adding new features, these new procedures would be implemented have taken the name "system calls". A system call is a mechanism whose name and number in unique, which is responsible for invoking a process to request a service to the operating system, hence, the user requests specifics and new tasks by a user program and this is performing executes the system call to allocate the resources available and needed. The implementation of system calls in a new Linux kernel through a compilation of it is a very easy process, but nevertheless should not be minimized as a simple task, because it must take into account knowledge of C programming language, management libraries, file manipulation process configuration files to support the boot process, the aim is that, when you start your operating system work in a personalized environment with new functionality. Companies dedicated to solving computer problems, can work not only in the area of software development but in terms of operating systems personified for users, allowing you to work in new environments with specific procedures and better use of resources areas.

**Keywords:** System call, implement, kernel, free software, Linux.

## ÍNDICE DE CONTENIDOS

PORTADA O CARÁTULA.....	i
CESIÓN DE DERECHOS DE AUTOR .....	ii
RESUMEN.....	iii
ABSTRACT.....	iv
ÍNDICE DE CONTENIDOS.....	v
ÍNDICE DE TABLAS.....	vi
ÍNDICE DE FIGURAS.....	vi
1 INTRODUCCION.....	1
1.1 MARCO CONTEXTUAL .....	2
1.2 PROBLEMA.....	2
1.3 OBJETIVO GENERAL.....	2
2 DESARROLLO .....	2
2.1 MARCO TEÓRICO .....	2
CentOS Linux .....	2
El Kernel .....	3
Versiones del kernel de Linux.....	3
Llamadas al Sistema.....	3
Comunicación con el kernel.....	3
El acceso a la llamada al sistema desde el espacio de usuario .....	4
2.2 MARCO METODOLÓGICO.....	4
2.3 RESULTADOS .....	7
3 CONCLUSIONES .....	10
REFERENCIAS BIBLIOGRAFICAS .....	11
ANEXOS.....	12
REACTIVO PRÁCTICO .....	12
REPORTE DE ANÁLISIS EN EL SISTEMA URKUND .....	13

## ÍNDICE DE TABLAS

Tabla 1. Ejemplo de versiones del Kernel de CentOS .....	3
--	---

## ÍNDICE DE FIGURAS

Gráfico 1. Relación entre Usuario, Kernel y Hardware .....	5
Gráfico 2. Invocación de una Llamada al Sistema .....	6
Gráfico 3. Ventana de Inicio de CentOS .....	6
Gráfico 4. Diagrama de Flujo - Llamada al Sistema 337.....	7
Gráfico 5. Diagrama de Flujo - Programa de Usuario .....	8
Gráfico 6. Listado de Mensajes del Kernel .....	9
Gráfico 7. Listado de Mensajes del Kernel, Mensaje de Ejecución de Nuevas Llamadas al Sistema.....	9

## 1 INTRODUCCION

Bovet, Daniel & Cesati, Marco (2006) expresan que los sistemas operativos Linux, por tratarse de sistemas de código libre nos permiten crear modificaciones de los mismos, este proceso se lo puede realizar mediante la compilación del kernel, en cuyo caso nos permitirá implementar nuevas llamadas al sistema para ejecutar nuevas funcionalidades.

Las llamadas al sistema nos permiten interactuar directamente con el kernel, el mismo que está en constante relación con los dispositivos de hardware, el crear nuevas llamadas permite un mejor manejo de recursos y crear sistemas personalizados, teniendo en consideración las necesidades de los usuarios.

Una de las ventajas de implementar nuevas llamadas al sistema es, que estas funciones permiten manejar procesos, señales, archivos, datos, tiempo, etc.

Durante la configuración para el proceso de compilación del nuevo kernel se están creando los archivos, dependencias, módulos entre otros desde cero, por lo cual es necesario añadir los archivos nuevos de extensión .c en el kernel, para que al compilar se creen conjuntamente los objetos de dichos archivos. Teniendo listo el nuevo kernel se podrá comprobar el funcionamiento de las llamadas mediante programas de usuario, en los cuales solo se deberá declarar las librerías necesarias para realizar la invocación y que se ejecute con éxito.

Trabajar directamente con el kernel de una distribución Linux, es una tarea fácil pero eso no significa que no es considerada de índole profesional, ya que existe una manipulación de los archivos de configuración del sistema operativo, se deberá tener un completo dominio de comandos, conceptos, habilidades para dinamizar el mejor desempeño de las herramientas y permitir que se desarrollen nuevos sistemas, personalizando los entornos de trabajo de los clientes.

Día a día, se van desarrollando nuevas tecnologías y así, las necesidades de los usuarios se vuelven más específicas y complejas para automatizar toda clase de procedimientos; al desarrollar nuevas funciones permitimos que el desempeño de las actividades de una empresa o institución vaya proyectándose al desarrollo de la misma y por ende de la sociedad. La implementación de nuevas llamadas al sistema podría ser considerado en el ámbito de desarrollo de nuevos proyectos informáticos trabajando en plataformas de código libre, flexibles y que se ajustarían a los requerimientos que se le programen y configuren.

## **1.1 MARCO CONTEXTUAL**

La compañía ABC es una empresa dedicada al desarrollo de software, su sede es en el parque industrial de Cuenca. Esta empresa tiene una participación en el mercado de desarrollo de software de gestión en un 55%, siendo libre en el desarrollo de diversos aplicativos para empresas importantes en la región. En la actualidad adelanta un proyecto sobre el desarrollo de un sistema operativo basado en el kernel de Linux y aparte de los componentes comunes, necesitan implementar nuevas funcionalidades de acuerdo a su proyecto. La necesidad de mejorar día a día ha permitido que los desarrolladores migren a nuevas formas de trabajar con los sistemas y no solo trabajar en áreas como programación, bases de datos o redes, sino, con los sistemas operativos, mejorándolos, modificándolos adaptándolos a las necesidades de los usuarios. Las llamadas al sistema, permiten implementar nuevas funcionalidades mediante la manipulación directa con el kernel, personalizar los entornos de trabajo y darles un enfoque más sofisticado y pensando en sus necesidades y comodidad.

## **1.2 PROBLEMA**

¿De qué forma se podría implementar nuevas funcionalidades a un sistema operativo, dándole un entorno personalizado, cumplir procesos específicos y garantizar el mejor uso de recursos?

## **1.3 OBJETIVO GENERAL**

Implementar una nueva funcionalidad al kernel de la distribución de Linux CentOS mediante llamadas al sistema, para evidenciar las ventajas de crear nuevas funciones.

## **2 DESARROLLO**

### **2.1 MARCO TEÓRICO**

#### **CentOS Linux**

Del inglés Community Enterprise Operating System. Es una distribución Linux, cuyo sistema operativo es de clase empresarial y libre. CentOS tiene muchas ventajas ya que se lo ha definido como fácil de instalar, pensando en el usuario, y brindar estabilidad. (Boronczyk & Negus, 2009)



## El Kernel

El kernel también llamado núcleo es una de las partes esenciales del sistema operativo, ya que esta directamente en contacto con los componentes de la maquina es decir el hardware, siendo el unico capaz de poder manipularlos.

Cuando el hardware quiere comunicarse con el sistema, emite una interrupción que literalmente intercepta las actividades del procesador, que a su vez interrumpe el al kernel. Las interrupciones están identificadas, por lo tanto el kernel utilizará este número para ejecutar el controlador específico de la interrupcion para responder acertadamente. Como expresa Robert Love, podemos manifestar que *“la interfaz de usuario es la porción más externa del sistema operativo, el kernel o núcleo es la más interna.”* (Linux Kernel Development, 2010, pág. 4)

## Versiones del kernel de Linux

Es fundamental conocer que cada distribución de CentOS Linux trabaja con una versión de kernel diferente, por lo tanto es necesario que al momento de implementar un nuevo kernel se tome en consideración este factor.

**Tabla 1. Ejemplo de versiones del Kernel de CentOS**

CentOS versión	Arquitectura	Kernel versión
6.2	IA-32, x86-64	2.6.32-220
6.4	IA-32, x86-64	2.6.32-279
7.0	x86-64	3.10.0-123

**Fuente: CentOS Official Page**

## Llamadas al Sistema

En Linux, las llamadas o syscalls; son las interfaces emitidas por el kernel entre el modo usuario y el hardware que invocan un procedimiento y al mismo tiempo solicitan los servicios que ofrece el sistema operativo.

El usuario no conoce la manera en la que se invoca una llamada al sistema que ocurre durante la ejecución. Tan solo se ajusta a la salida de los resultados que el API se encarga de ejecutar. Los cuales son gestionados por la biblioteca de soporte durante el tiempo de ejecución. (Silberschatz, Baer Galuin, & Gagne, 2009)

## Comunicación con el kernel

Cuando el kernel implementa diferentes llamadas al sistema, el CPU cambia al modo kernel e inicia la ejecución de una función del núcleo. Este procedimiento se

genera mediante el salto a una función de lenguaje ensamblador llamado manejador de llamada al sistema (system call handler).

El proceso de modo usuario debe pasar un parámetro llamado número de llamada de sistema, que es una identificación única, para poder definir el requerimiento de la llamada.

Todas las llamadas al sistema retornan un valor entero. En el kernel el retorno del valor de 0 denota una exitosa terminación de la llamada al sistema, caso contrario, es porque se ha producido un error en el código de la llamada, ya sea por parámetros inválidos o error en el algoritmo y será retornada a la aplicación mediante la variable `errno`. (Bovet & Cesati, 2006)

## **El acceso a la llamada al sistema desde el espacio de usuario**

Por lo general, las librerías C proveen apoyo para las llamadas al sistema. En las aplicaciones de usuario se declaran en las cabeceras una referencia para que la llamada se realice con los parámetros de las funciones del programa (rutinas de biblioteca que usan las llamadas al sistema).

Además, debemos establecer el contenido del registro, que se nombran mediante `_syscalln()`, donde `n` es entre 0 y el número 6. Lo que corresponde con el número de parámetros con los que trabajará la llamada al sistema.

Además no debemos olvidar declarar el nombre de la llamada al sistema, el tipo y el nombre de cada parámetro que usaremos dentro de la función. Esta macro expande una función C con ensamblador, describiendo los pasos de la llamada al sistema y emitir la interrupción correspondiente para que se ejecute en el kernel y devolver la respuesta a la aplicación que se está ejecutando.

## **2.2 MARCO METODOLÓGICO**

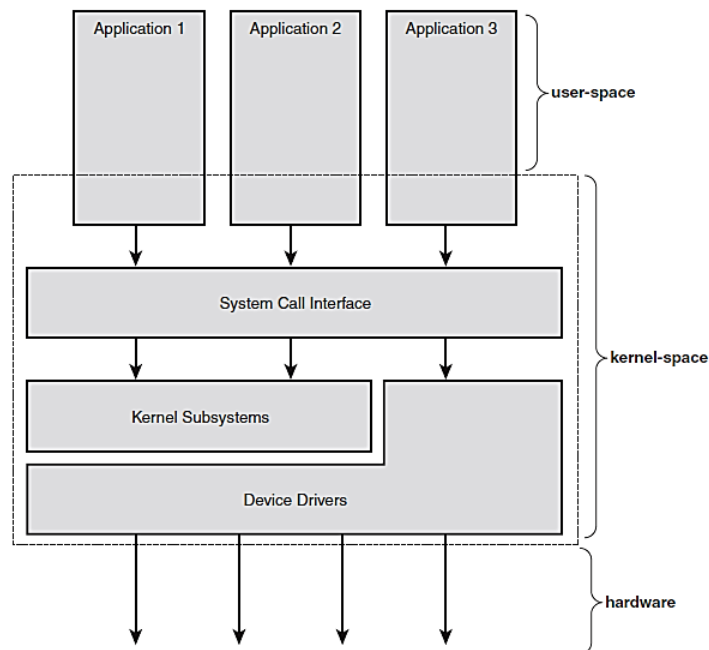
Para implementar una llamada al sistema es necesaria la configuración, programación y compilación de un kernel de una distribución CentOS Linux, se usó la versión 2.6.32.25 que es compatible con la versión de CentOS 6.2 que se ha instalado. Es indispensable la instalación de los paquetes `ncurses`, que se encargaran de proveer una API para escribir interfaces basadas en texto y `gcc`, que es un compilador de C de GNU; y por supuesto, descargar nuestro kernel en el que se trabajará para ser compilado, que se realiza en la página [www.kernel.org](http://www.kernel.org) y será necesario que sea descomprimido en la ruta `/usr/src`. Los sistemas operativos de todas las arquitecturas incluyendo los que soportan Linux, proporcionan el concepto de interrupciones.

Al momento de comunicar con el sistema, se emitió una interrupción en el procesador y a su vez al núcleo o kernel, en nuestro caso haremos referencia al uso del kernel 2.6.32.25. Al identificar dicha interrupción mediante la identificación de una llamada al sistema esta ejecuta un controlador específico para procesar y responder dicha interrupción.

Por eso al momento que se ejecuta una aplicación se genera una serie de eventos, tales como(Ver Gráfico 1):

- En el espacio de usuario, se ejecuta el código de usuario en un proceso.
- Mediante la ejecución del código se invoca una llamada al sistema.
- En el espacio kernel, se genera la ejecución del proceso de la llamada, como respuesta a la interrupción generada por la petición.
- Se ejecuta un controlador específico de un dispositivo de hardware.

**Gráfico 1. Relación entre Usuario, Kernel y Hardware**



**Fuente: Linux Kernel Development Third Edition (Robert Love - 2010)**

Una llamada al sistema se caracteriza por un nombre y un número único que la identifica. Este número puede encontrarse en el archivo `<asm/unistd.h>` ubicado en la ruta `/usr/src/linux-2.6.32.25/arch/x86/include/asm`

En los programas fuente de kernel, una llamada al sistema posee el siguiente prototipo, y los argumentos pueden considerarse entre 0 y 5. La palabra clave `asm linkage` es un macroinstrucción definida en el archivo `<linux/linkage.h>` ubicado en la ruta `/usr/src/linux-2.6.32.25/include/linux`

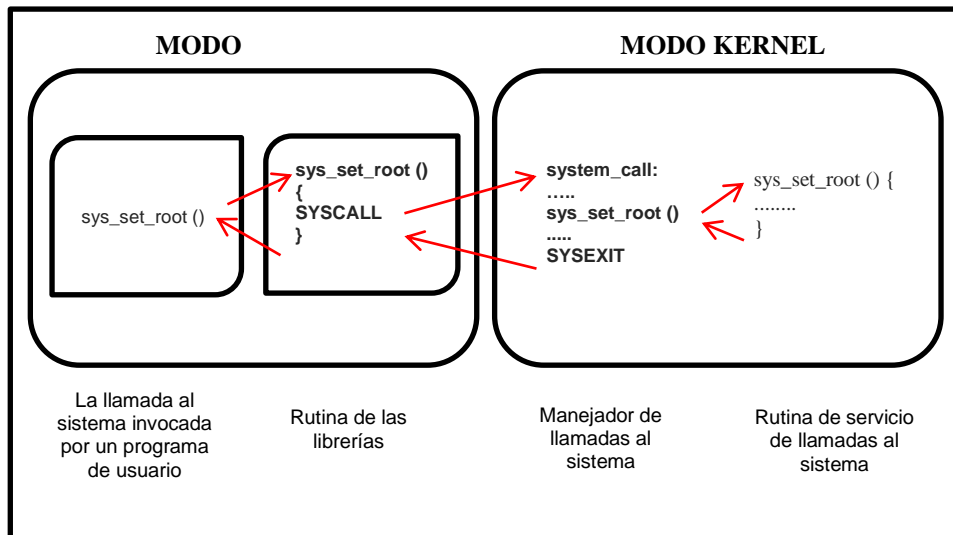
Cuando un proceso ejecuta una llamada al sistema, llama a la función correspondiente de la librería C. esta función trata los parámetros y hace pasar al proceso al modo kernel (Ver Gráfico 2).

El proceso es el siguiente:

- Los parámetros de la llamada al sistema se colocan en ciertos registros del procesador.
- Se provoca un bloqueo desencadenando la interrupción lógica 0x80.

- Este bloqueo proporciona el paso del proceso al modo kernel, el proceso ejecuta la función *system\_call*.
- Esta función utiliza el número de llamada al sistema como índice de la tabla *sys\_call\_table*, que contiene las direcciones de las llamadas al sistema y llama a la función del kernel correspondiente a la llamada del sistema.
- Al volver de esta función, *system\_call* vuelve a quien ha llamado, este entorno vuelve a pasar al proceso a modo usuario.

**Gráfico 2. Invocación de una Llamada al Sistema**



**Desarrollado por: Angie E. Parra Lambert**

**Adaptado de: The Linux Kernel Third Edition (Bovet & Cesati - 2006).**

Antes de compilar nuestro kernel es necesario que implementemos todas las llamadas que darán nuevas funcionalidades a nuestro sistema.

Creando un nuevo archivo de extensión .c en la extensión `/kernel-version/kernel`

Durante el proceso de compilación se irán incorporando estas llamadas como funciones propias de un sistema personalizado. (Ver Gráfico 3).

**Gráfico 3. Ventana de Inicio de CentOS**

```
GNU GRUB version 0.97 (639K lower / 924608K upper memory)
CentOS 6.2 - Actualizado - Angie Parra Lambert (kernel 2.6.32.25)
CentOS (2.6.32-220.el6.i686)
```

**Desarrollado por: Angie E. Parra Lambert.**

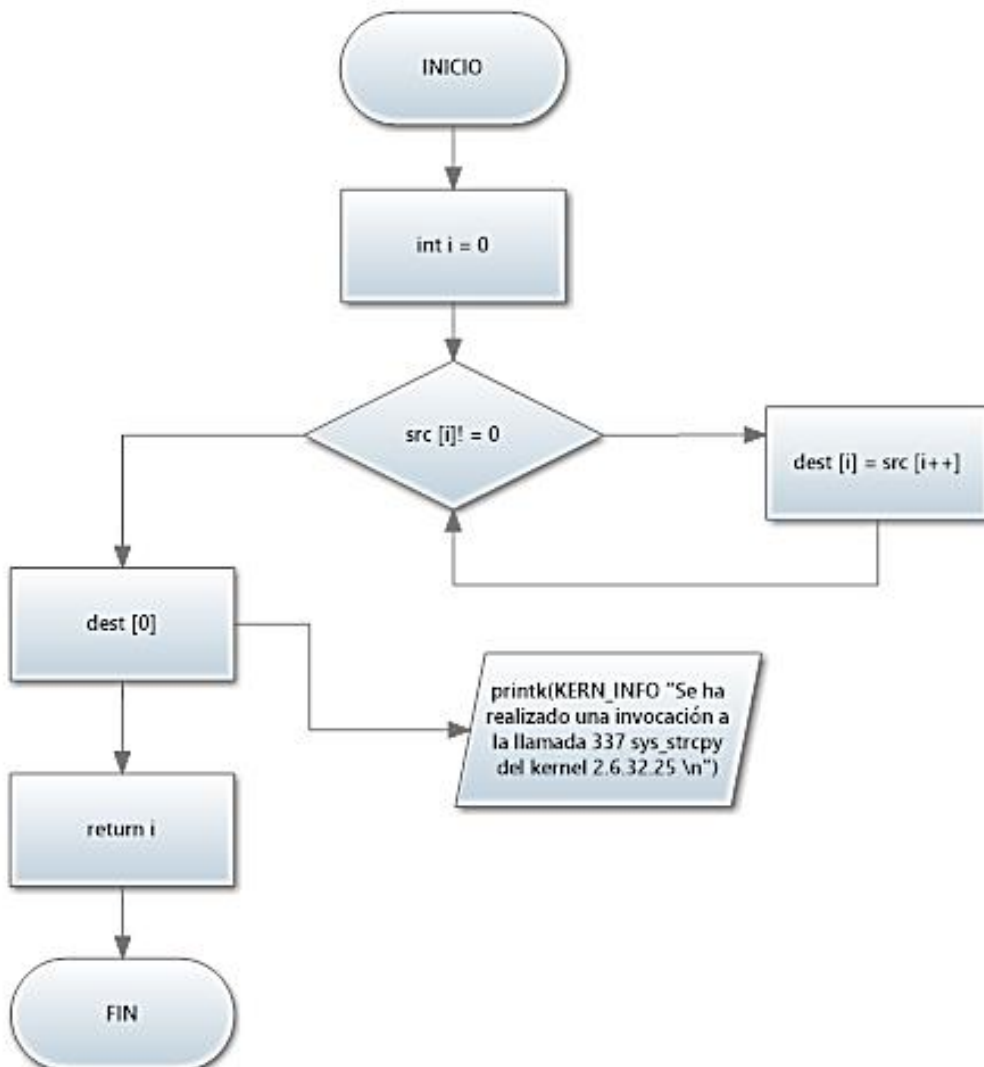
## 2.3 RESULTADOS

Cuando se ha realizado el proceso de compilación con éxito, al momento de verificar la versión del kernel que tiene nuestro sistema, aparecerá la del nuevo kernel instalado, configurado y compilado.

```
$ uname -r  
2.6.32.25
```

Para la prueba de la llamada al sistema por medio de un programa de usuario, se creó una función que permita copiar una cadena e caracteres de un valor ingresado por teclado. (Ver Gráfico 4)

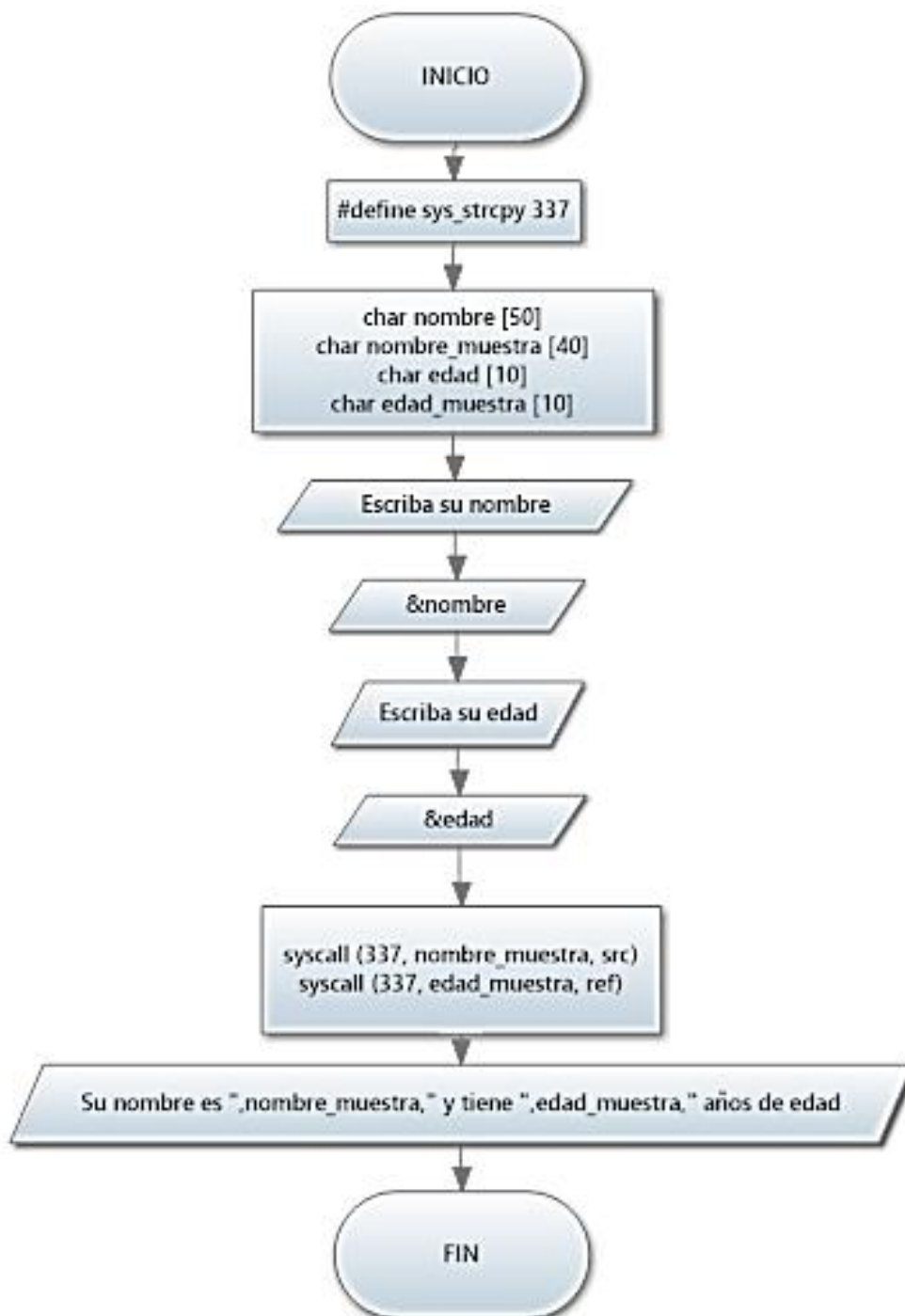
**Gráfico 4. Diagrama de Flujo - Llamada al Sistema 337**



**Desarrollado por: Angie E. Parra Lambert.**

Se creó un programa sencillo de usuario para realizar las pruebas de funcionamiento respectivas, para lo cual es fundamental declarar qué llamada al sistema que invocaremos, la cual se identificará por su número único que por medio de las librerías `#include <linux/syscall.h>` la cual contiene los parámetros de la llamada e `#include <linux/unistd.h>` donde se encuentra el listado de las mismas. (Ver Gráfico 5)

**Gráfico 5. Diagrama de Flujo - Programa de Usuario**



**Desarrollado por: Angie E. Parra Lambert.**

Una vez ejecutado el programa de usuario es necesario abrir un terminal y ejecutar el comando **dmesg**, que, al ser ejecutado, nos muestra en el terminal una lista con los mensajes del **kernel**. En caso de que la llamada al sistema no se haya ejecutado, ya sea por error de programación o compilación, tan solo aparecerán los mensajes por default del kernel. (Ver Gráfico 6)

#### **Gráfico 6. Listado de Mensajes del Kernel**

```
SELinux: initialized (dev rpc_pipefs, type rpc_pipefs), uses genfs_contexts
SELinux: initialized (dev autofs, type autofs), uses genfs_contexts
SELinux: initialized (dev autofs, type autofs), uses genfs_contexts
SELinux: initialized (dev autofs, type autofs), uses genfs_contexts
eth0: no IPv6 routers present
fuse init (API version 7.13)
[root@localhost Desktop]# █
```

**Desarrollado por: Angie E. Parra Lambert.**

Caso contrario, si la llamada al sistema ha sido ejecutada con éxito, el listado de los mensajes del kernel se incrementarán y aparecerá el que se ha asignado dentro de la llamada que hemos programado. (Ver Gráfico 7)

#### **Gráfico 7. Listado de Mensajes del Kernel, Mensaje de Ejecución de Nuevas LLamadas al Sistema**

```
SELinux: initialized (dev rpc_pipefs, type rpc_pipefs), uses genfs_contexts
SELinux: initialized (dev autofs, type autofs), uses genfs_contexts
SELinux: initialized (dev autofs, type autofs), uses genfs_contexts
SELinux: initialized (dev autofs, type autofs), uses genfs_contexts
eth0: no IPv6 routers present
fuse init (API version 7.13)
Se ha realizado una invocacion a la llamada 337 sys_strcpy del kernel 2.6.32.25
Se ha realizado una invocacion a la llamada 337 sys_strcpy del kernel 2.6.32.25
[root@localhost Desktop]# █
```

**Desarrollado por: Angie E. Parra Lambert.**

### 3 CONCLUSIONES

El núcleo de Linux implementa las llamadas al sistema y la cadena de acontecimientos necesarios para ejecutar un proceso que se encuentra custodiado y será ejecutado en el kernel, transmitiendo el número del sistema y los argumentos respectivos. El proceso de crear y ejecutar un ejemplo del uso de una nueva llamada al sistema desde el espacio de usuario es muy sencillo, así como también la compilación de un nuevo kernel, aun así, este proceso no debe ser subestimado, ya que con los conocimientos necesarios de comandos, configuración y manipulación de archivos de la distribución CentOS, se logrará implementar nuevas funcionalidades a un sistema operativo. Los ingenieros en sistemas ya no se están centrando específicamente al área de desarrollo de software, redes o área técnica y de mantenimiento, con la facilidad de trabajar con código libre permite que se generen nuevas perspectivas al momento de ofrecer productos personalizados para los usuarios, donde trabajen según sus necesidades aprovechando recursos y dedicados específicamente a funcionalizar procesos que no estaban incorporados en los entornos de trabajo de los sistemas operativos estándar y de uso común.

Existe una ligera discusión entre las ventajas y desventajas de compilar un kernel e implementar nuevas llamadas al sistema. Entre las ventajas se puede manifestar existirá una mayor eficiencia, ya que solo se cargaran los componentes que el usuario necesitará y permitir un mayor uso de recursos como memoria, aun así, existes ligeras contraindicaciones ya que el nuevo kernel no recibirá actualizaciones, ni soporte técnico, pero como tratarse de un kernel personalizado el administrador del sistema podrá proveerlos por su cuenta sin ningún problema y claro está, en caso de olvidar alguna configuración será necesario una re compilación.

Aun así, se evidenció una nueva forma de crear entornos de trabajo personalizados, creando funcionalidades y garantizando el desempeño óptimo de los procesos.



## REFERENCIAS BIBLIOGRAFICAS

- Boronczyk, T., & Negus, C. (2009). *CentOS Bible* (First ed.). Indianapolis, Indiana, United States of America: Wiley Publishing Inc.
- Bovet, D., & Cesati, M. (2006). *Understanding The Linux Kernel* (Third ed.). Los Angeles, California, United States of America: O' Reilly Media, Inc.
- CentOS Linux. (2015). *CentOS*. Obtenido de [www.centos.org](http://www.centos.org)
- Edgardo E. , H., Julio A., B., & Franco M., L. (2009). *Programación de Controladores para Linux*. Universidad Nacional de Córdoba, Facultad de Matemática, Astronomía y Física, Córdoba, Argentina.
- García Cortés , R. (2013). *Reporte de nueva Llamada al sistema al Kernel 3.11.8*. México.
- García Gómez, R. J., Sánchez Esteban, J. J., & Vivas Rebuella, J. M. (2005-2006). *Emulador de llamadas al sistema de Linux bajo arquitectura PowerPC*. Universidad Complutense de Madrid, Facultad de Informática, Madrid.
- Love, R. (2010). *Linux Kernel Development* (Third ed.). Crawfordsville, Indiana, United States of America: Pearson Education Inc.
- Silberschatz, A., Baer Galuin, P., & Gagne, G. (2009). *Operating System Concepts* (Eighth ed.). Jefferson City, United States of America: Jhon Wiley & Sons Inc.

## **ANEXOS**

### **REACTIVO PRÁCTICO**

La compañía ABC es una empresa dedicada al desarrollo de software, su sede es en el parque industrial de Cuenca. Esta empresa tiene una participación en el mercado de desarrollo de software de gestión de un 55%, siendo líder en el desarrollo de diversos aplicativos para empresas importantes de la región. En la actualidad adelanta un proyecto sobre el desarrollo de un sistema operativo basado en el kernel de Linux y aparte de los componentes comunes necesitan implementar nuevas funcionalidades acorde su proyecto. Añadir al kernel de una distribución de Centos Linux, una nueva llamada al sistema cuya funcionalidad no exista. Demuestre la nueva funcionalidad del sistema a través de un programa de usuario cualquiera.

## Urkund Analysis Result

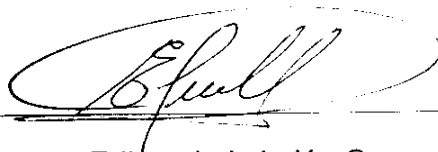
**Analysed Document:** Informe de Parte Practica - Angie Estefany Parra Lambert.docx  
(D16312046)  
**Submitted:** 2015-11-21 01:51:00  
**Submitted By:** angie.pl-loverock@hotmail.com  
**Significance:** 4 %

### Sources included in the report:

[http://www.ual.es/~acorral/DSO/Tema\\_1.pdf](http://www.ual.es/~acorral/DSO/Tema_1.pdf)  
<https://www.yumpu.com/es/document/view/22938779/visian-general-e-introduccian-al-kernel- universidad-de-almera-a/25>

### Instances where selected sources appear:

3



Ing. Edison Luis Loján Cueva  
C.I. 0703249698

**RESPONSABLE DEL CASO PRÁCTICO**