



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
IC	2003-1	5314	Sistemas Operativos

PRÁCTICA No.	LABORATORIO DE		DURACIÓN (HORA)
5	NOMBRE DE LA PRÁCTICA	Comunicación entre procesos "Envío de señales"	2

1. INTRODUCCIÓN

La comunicación entre procesos, en inglés **IPC** (*Inter-process Communication*) es una función básica de los sistemas operativos. En esta práctica se trabajará con un tipo especial de comunicación entre procesos llamada Envío de señales.

2. OBJETIVO (COMPETENCIA)

En esta práctica el alumno se familiarizará con la comunicación entre procesos, para cual enviará señales de un proceso a otro utilizando la función kill().

3. FUNDAMENTO

Aunque muchas tareas pueden realizarse en procesos aislados, la gran mayoría requieren la intervención de más de un proceso. Para que dichos procesos cooperantes lleven a buen término una tarea común, es necesario algún tipo de comunicación entre ellos. Los mecanismos de comunicación entre procesos (IPC's) habilitan mecanismos para que los procesos puedan intercambiar datos y sincronizarse. A la hora de comunicar dos procesos, se consideran dos situaciones diferentes:

1. Que los procesos se estén ejecutando en una misma máquina, o bien
2. Que los procesos se ejecuten en máquinas diferentes.

En el caso de comunicación local entre procesos, aunque existen diferentes mecanismos que se engloban bajo esta denominación común, cada uno de ellos tiene su propósito específico.

Así, las señales son interrupciones software y, como tales, no deberían considerarse parte de la forma habitual de comunicar procesos y su uso debería restringirse a la comunicación de eventos o situaciones excepcionales. Para

Formuló MC Alicia del R. López Aguirre	Revisó MC Gloria E. Chávez Valenzuela	Aprobó	Autorizó Dr Maximiliano de las Fuentes Lara
Maestro	Coordinador de Programa Educativo	Gestión de Calidad	Director de la Facultad



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formatos para prácticas de laboratorio

dar solución a necesidades de comunicación más genéricas entre procesos, Unix y su adaptación para PC (Linux) utilizan diferentes soluciones que emplean como canal la memoria principal.

Una de las primeras formas de este tipo de comunicación son los *pipes* o tuberías. Los *pipes* permiten un mecanismo de comunicación unidireccional sencillo, pero también limitado. Como soluciones más elaboradas a la comunicación entre procesos se incorporan otros tres tipos de mecanismos: semáforos, memoria compartida y colas de mensajes.

Tipos de señales

Todas las señales tienen una rutina de servicio, o una acción, por defecto. Dicha acción por defecto puede ser de varios tipos:

- **A:** terminación del proceso.
- **B:** ignorar la señal.
- **C:** volcar un farchivo *core*.
- **D:** parar el proceso.
- **E:** la rutina de servicio no puede modificarse.
- **F:** la señal no puede ser ignorada.

Cada una de las señales se identifica mediante un número entero mayor que cero. En la tabla 1 se describen las señales más importantes.

Señal	Identificador	Tipo	Descripción
SIGKILL	9	AEF	Aborta la ejecución del proceso.
SIGSTOP	19	DEF	Para el proceso.
SIGCONT	18		Continúa si el proceso está parado.
SIGCHLD	17	B	Un proceso hijo ha terminado o ha parado.
SIGUSR1	10	A	Señal de usuario.
SIGUSR2	12	A	Señal de usuario.

Tabla 1 Resumen de las principales señales

Nota: Para obtener más información, se recomienda ejecutar en un terminal: **man 7 signal**



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formatos para prácticas de laboratorio

C bajo Linux proporciona una serie de funciones para llamadas al sistema que nos permiten el envío y recepción de señales, entre ellas tenemos la función `kill()`. A continuación se explica como es que esta función trabaja y se muestra un ejemplo de ella.

Envío de una señal utilizando la función `kill()`

Cabecera:

```
#include <sys/types.h>
#include <signal.h>
```

```
int kill(int pid, int sig);
```

Comportamiento de la llamada:

`kill()` se utiliza para enviar una señal concreta, *sig*, a un proceso determinado que se caracteriza por su identificador, *pid*. Si se produjo cualquier error devolverá un valor **-1** y, en caso contrario, devuelve el valor **0**.

Ejemplo:

En el siguiente código se muestra un ejemplo en el que un proceso envía la señal SIGSTOP a otro proceso cuyo **pid** recibe como parámetro.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <signal.h>

int main(int argc, char *argv[]) {

    int pid = 0;

    if (argc != 2) {
        printf(`usage: ejemplo_kill pid\n");
        exit(1);
    }
    pid = atoi(argv[1]);
    if (kill(pid, SIGSTOP) == -1) {
        printf(`\nError al enviar la señal SIGSTOP al proceso %d\n",pid);
        exit(1);
    }
    exit(0);
}
```



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

4. PROCEDIMIENTO (DESCRIPCIÓN)

A) EQUIPO NECESARIO

Computadora con sistema operativo Linux.

El alumno elaborará un programa que permita leer el PID de un proceso, posterior a esta lectura imprimirá el siguiente menú de opciones:

```

Parar proceso..... 1
Continuar proceso..... 2
Terminar proceso..... 3
Matar proceso..... 4
Salir..... 5
  
```

Práctica 5 del Manual de sistemas operativos
Para el plan de estudios 2003-1

Si el usuario selecciona la opción uno, dos, tres o cuatro deberá enviar la señal adecuada utilizando la función **kill()** al proceso indicado. Si el usuario selecciona la opción cinco el programa deberá terminar de la siguiente manera:

- Obtener el PID del proceso con la función **getpid()**
- Enviarle una señal al proceso mediante la función **kill()** para matar dicho proceso, es decir SUICIDAR AL PROCESO.

C)

CÁLCULOS Y REPORTE

5. RESULTADOS Y CONCLUSIONES

Al finalizar la práctica el alumno demostrará su habilidad para comunicar procesos mediante la función **kill()**

6. ANEXOS

7. REFERENCIAS

<http://148.231.82.20/~so>