SISTEMAS OPERATIVOS

UNIDAD 5

SISTEMAS DE LLAMADAS Y SEÑALES

La Victoria de la búsqueda, es la búsqueda en si misma.
(Paul Auster)

SYSTEM CALL

CONJUNTO DE FUNCIONES QUE PERMITEN LA GESTION DE LOS SERVICIOS EN MODO KERNEL POR PARTE DE LOS PEDIDOS QUE SE REALIZAN DESDE EL MODO USUARIO.

PROCESO

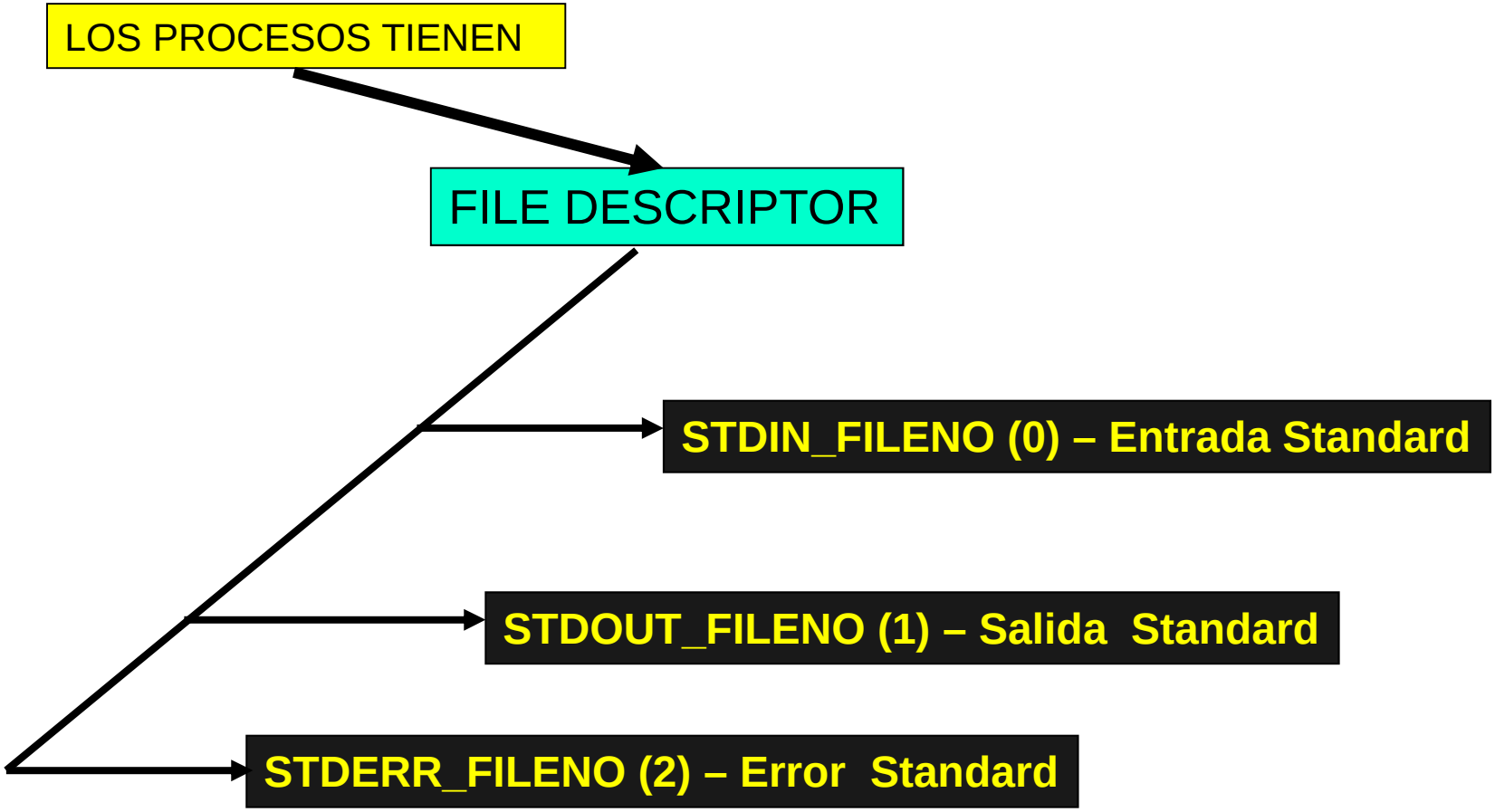
ATRIBUTOS

- PID (ID del proceso)
- PPID (ID del Padre)
- UID (ID del Usuario Real)
- EUID (ID del Usuario Efectivo)
- GID (ID del Grupo Real)
- EGIT (ID del Grupo Efectivo)

SYSTEM CALL

FORMATO GENERAL:
CREAT(2)
crear un archivo nuevo.
SYNOPSIS
include <sys/types.h>
include <sys/stat.h>
include <fcntl.h>
int creat (const char *pathname, mode_t mode);
DESCRIPTION
La función Creat crea un archivo regular nuevo, o prepara para volver a escribir uno ya existente.
ERRORS
[ENOSPC] no hay suficiente espacio en el sistema de archivos.
[EACCES] no se tiene permiso de acceso.
RETURN VALUE
Si no hay error retorna el file descriptor, si hay error retorna -1, y se guarda en la variable errno la causa del error.
SEE ALSO
chmod(2), close,…

SYSTEM CALL


LOS PROCESOS TIENEN
FILE DESCRIPTOR
STDIN_FILENO (0) – Entrada Standard
STDOUT_FILENO (1) – Salida Standard
STDERR_FILENO (2) – Error Standard

SYSTEM CALL

SYSTEM CALL PARA ACCESO A ARCHIVOS

open ( ) – Abrir archivo
read ( ) – Lectura archivo
close ( ) – Cerrar archivo
lseek ( ) – Posicionar un archivo
unlink ( ) – Destrucción enlace a directorio

SYSTEM CALL PARA CONTROL DE PROCESOS

exec ( ) – Ejecuta un proceso
main ( ) – Recibe argumentos desde linea de ejecución
fork ( ) – Crea un proceso
wait ( ) – Espera fin de un proceso
exit ( ) – Fin de un proceso

SYSTEM CALL

SYSTEM CALL PARA COMUNICACION ENTRE ARCHIVOS

dup ( ) – Redirección de mensajes
dup2 ( ) – idem.
pipe ( ) – Idem IPC

OTROS SYSTEM CALL

time ( ) – Retorna valor en sg desde 1/1/1970
gettimeofday ( ) – Hora actual, resolución en microsg
perror ( ) – Muestra info sobre error en proceso
chdir ( ) – Cambio de directorio de trabajo
getenv ( ) – Acceso al valor de una variable de entorno del proceso
setenv ( ) – Cambio en variables de entorno.
exit ( ) – Fin de un proceso

SYSTEM CALL

SYSTEM CALL PARA TRANSFERIR SEÑALES

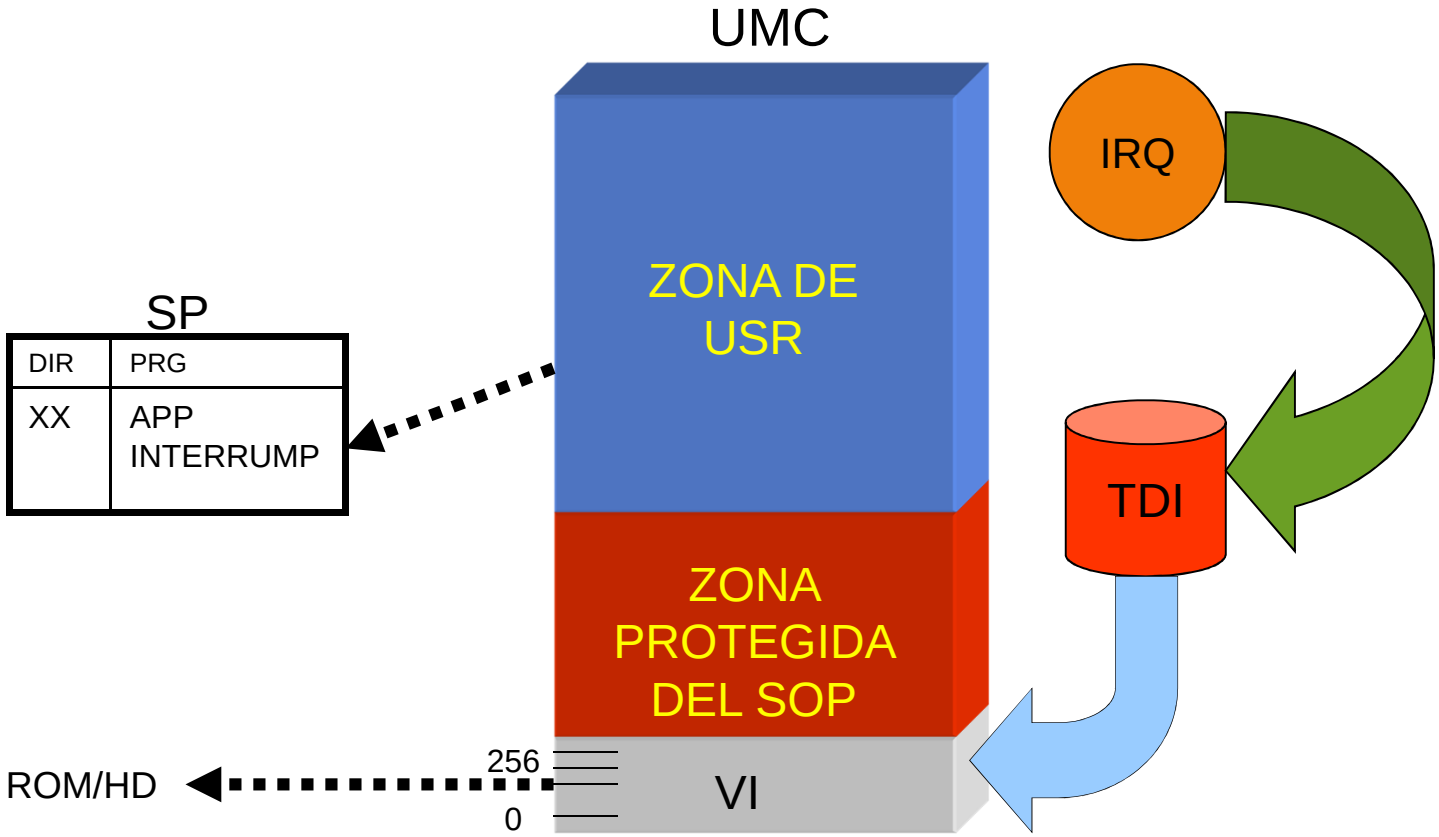
signal ( ) – usa signal.h

LISTADO DE SEÑALES

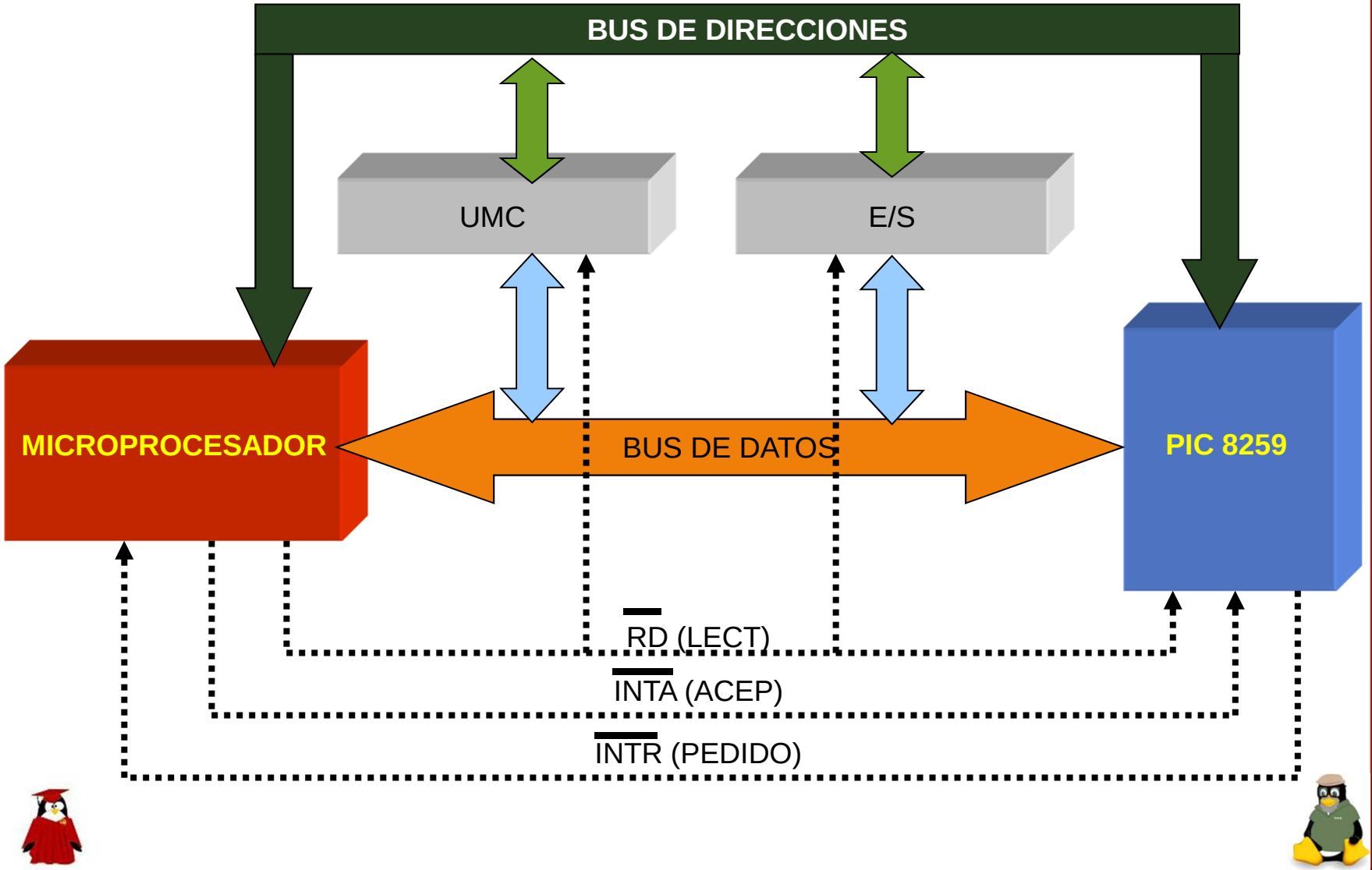
SIGHUP – Colgar. Desconexión de terminal
SIGINT – Interrupción. Generada por el teclado
SIGILL – Instrucción ilegal. No capturable
SIGFPE - Excepción aritmética, de coma flotante o división por cero
SIGKILL – Matar proceso. No se captura ni se ignora.
SIGBUS – Error en el Bus
SIGSEGV – Violación de Segmentación.
SIGPIPE – Escritura en un pipe para el cual no hay lectores.
SIGALRM – Alarma de reloj
SIGTERM – Terminación de un programa.

INTERRUPCIONES

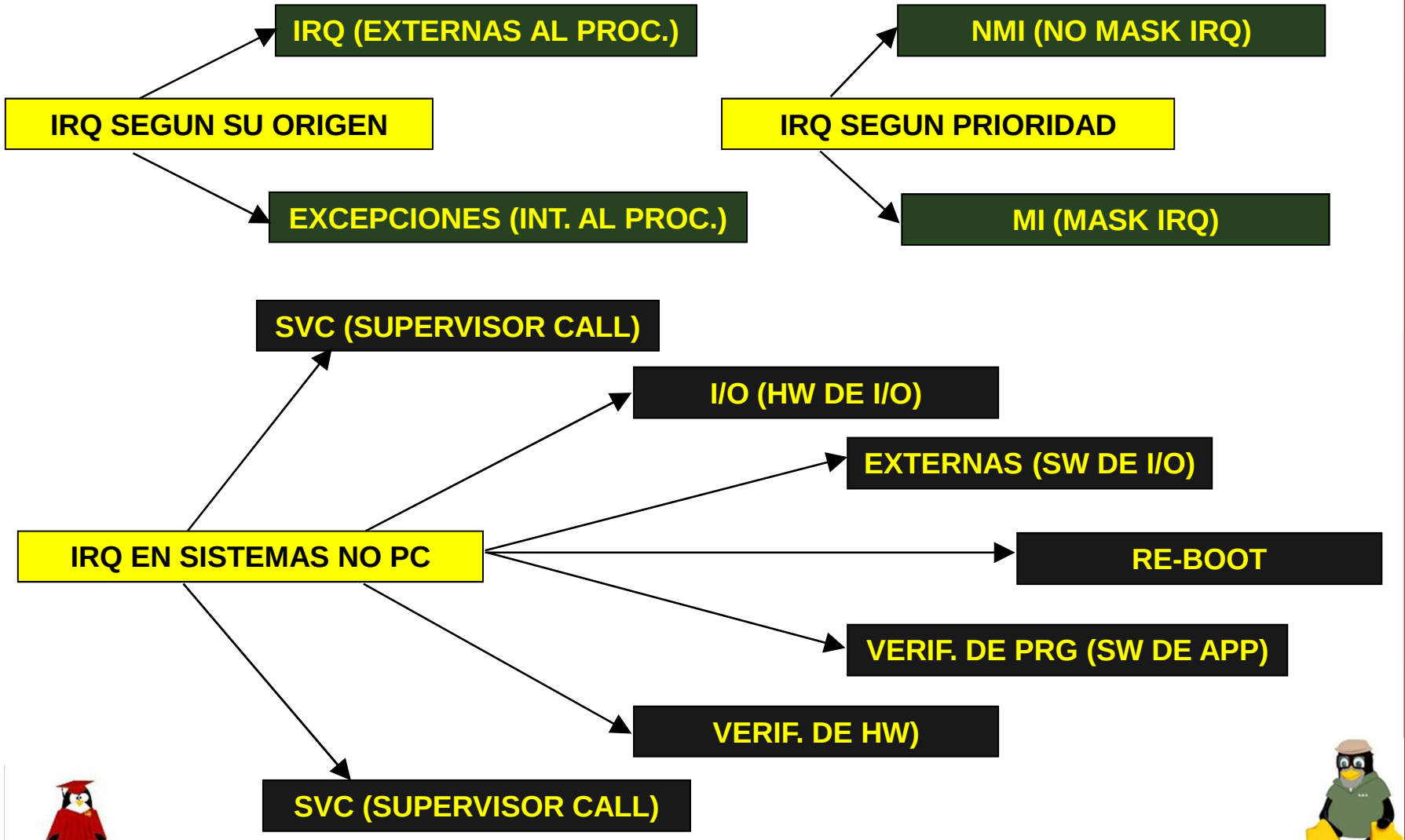
EVENTO QUE ALTERA LA FRECUENCIA/SECUENCIA DE EJECUCION DE INSTRUCCION



INTERRUPCIONES POR HW



INTERRUPCIONES: CLASIFICACION



INTERRUPCIONES: ALGORITMO FUNCIONAL

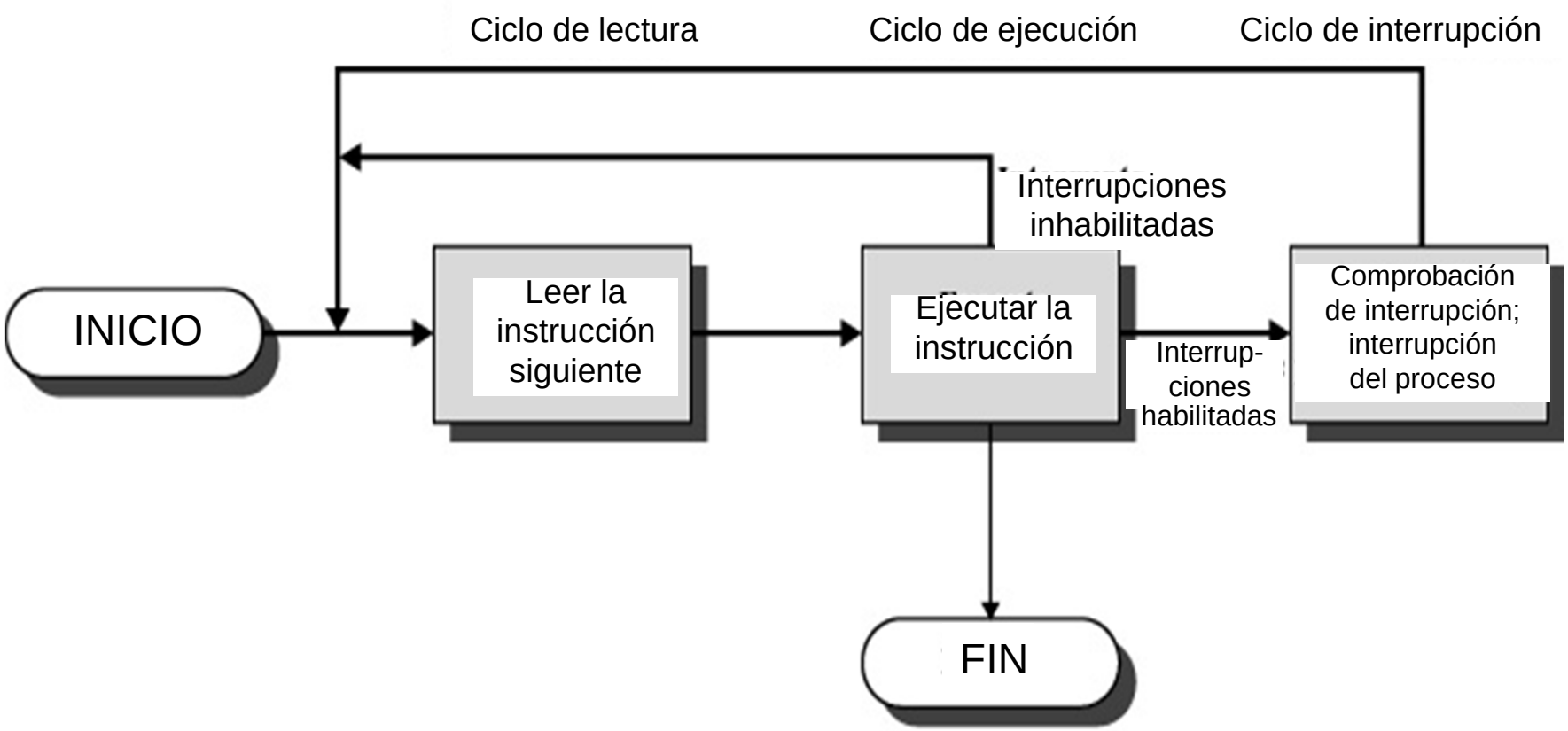
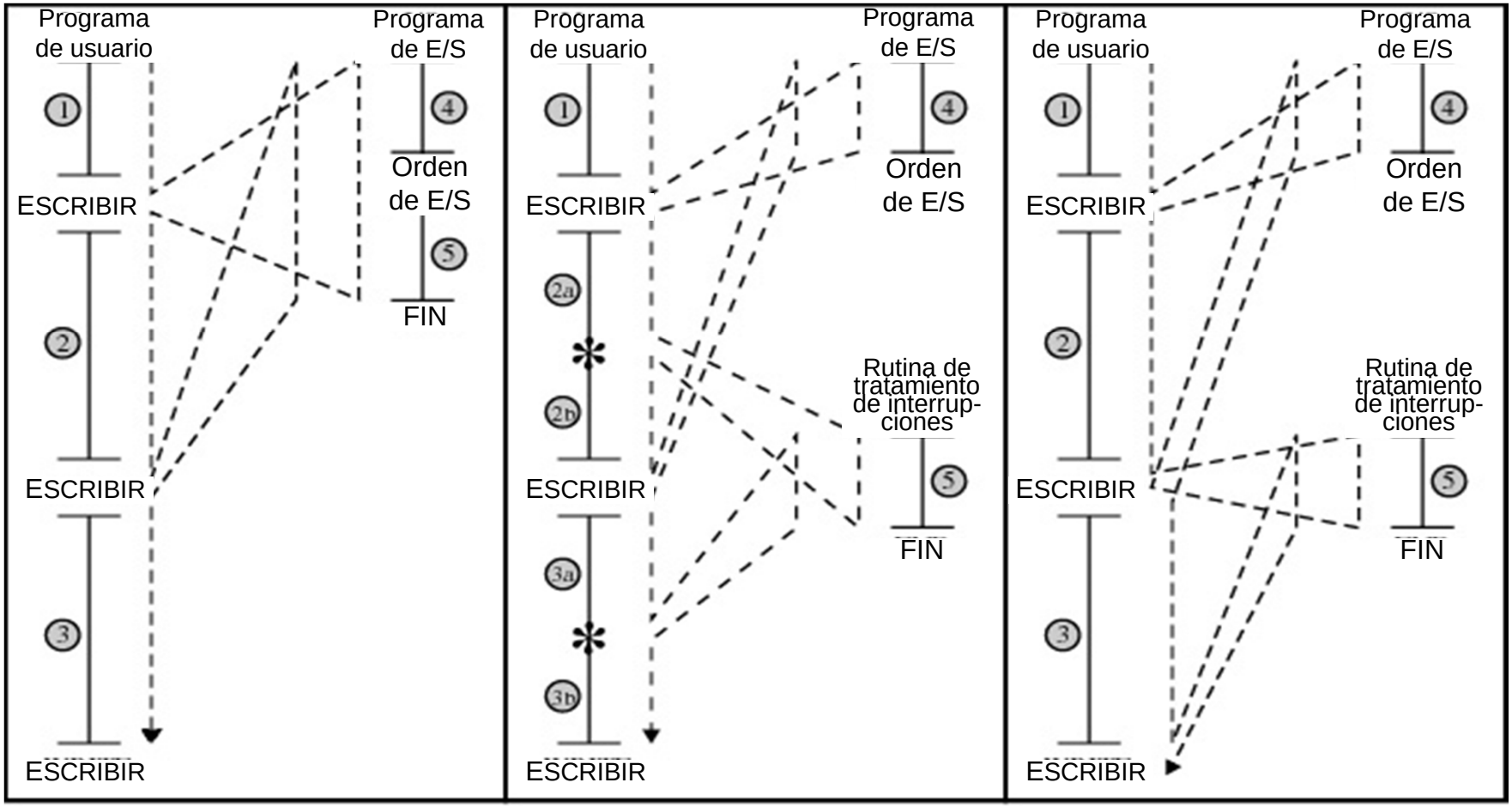


Figura 1.7. Ciclo de instrucción con interrupciones.

Williams Stallings SISTEMAS OPERATIVOS. Principios de interioridades. 4ta ed. Pearson Eduación S.A. Madrid, 200: 5-3177-4

INTERRUPCIONES: SECUENCIA DE EJECUCION



(a) Sin interrupciones

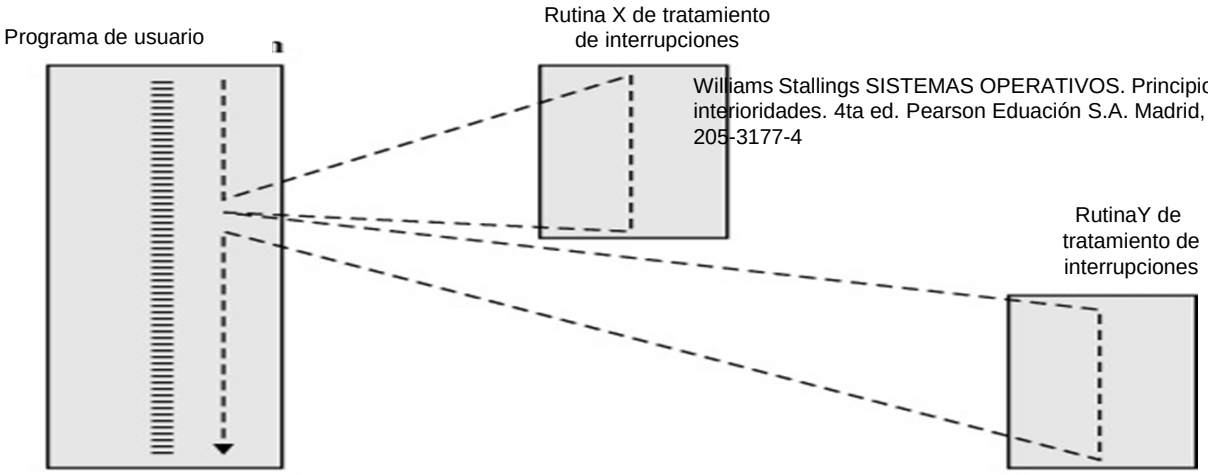
(b) Con interrupciones y corta espera de E/S

(c) Con interrupciones y larga espera de E/S

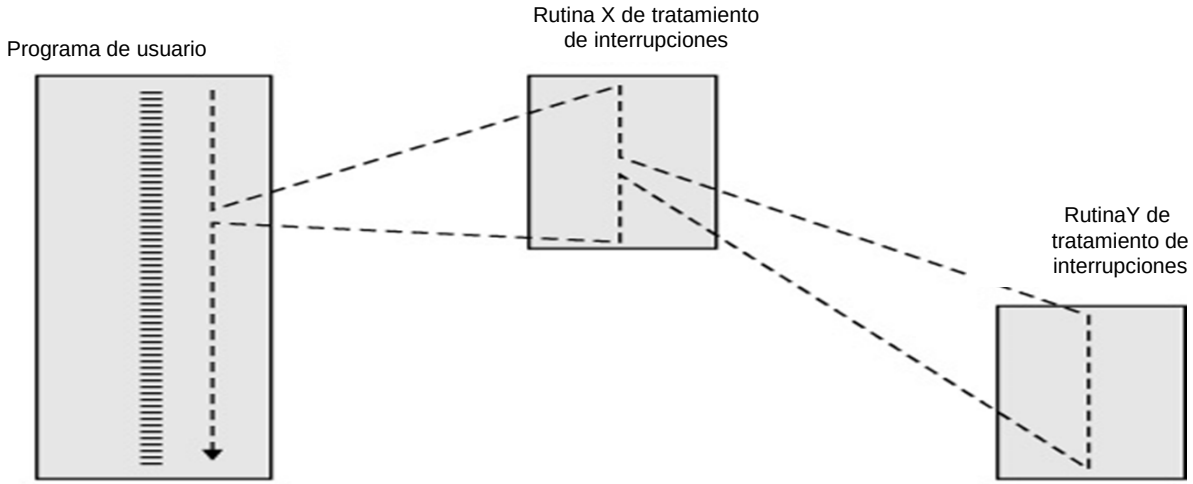
Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Eduación S.A. Madrid, 2001 ISBN: 84-205-3177-4

Figura 1.5. Flujo de control del programa con y sin interrupciones.

INTERRUPCIONES MULTIPLES



(a) Tratamiento secuencial de interrupciones

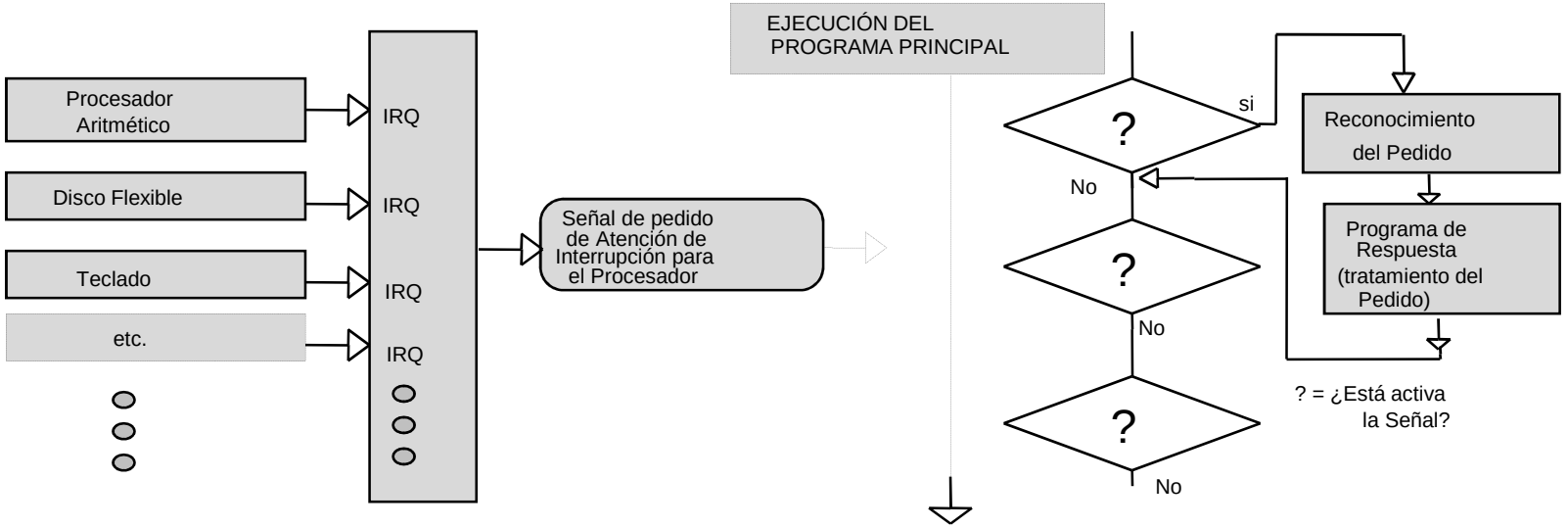


(b) Tratamiento de control con múltiples interrupciones

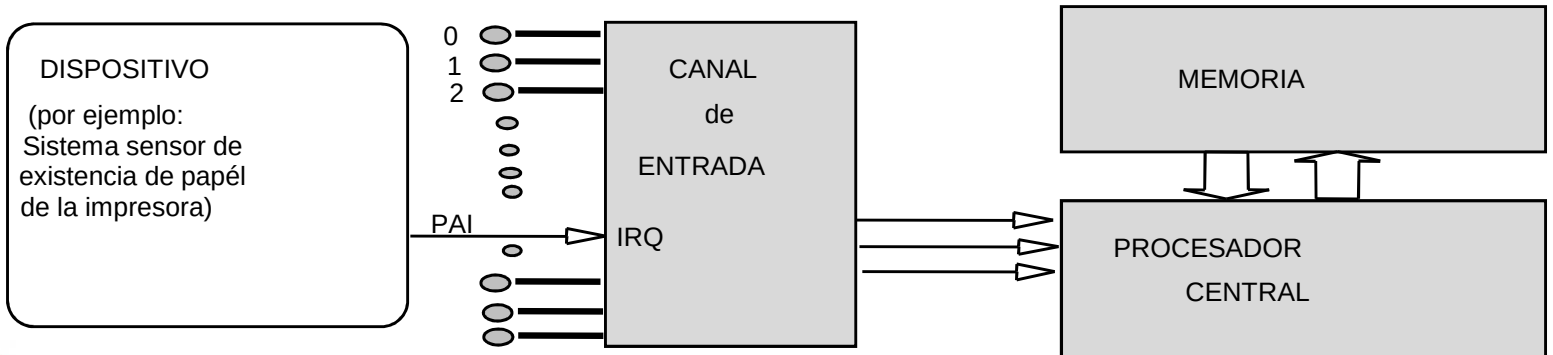
Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Eduación S.A. Madrid, 2001 ISBN: 84-205-3177-4

Figura 1.12. Transferencia de control con múltiples interrupciones.

IRQs LINUX



Pedido y satisfacción de IRQ



PAI = Pedido de Atención de Interrupción
Ej.: 0 = Hay Papél
1 = No Hay Papél

FIG. A

IRQs LINUX

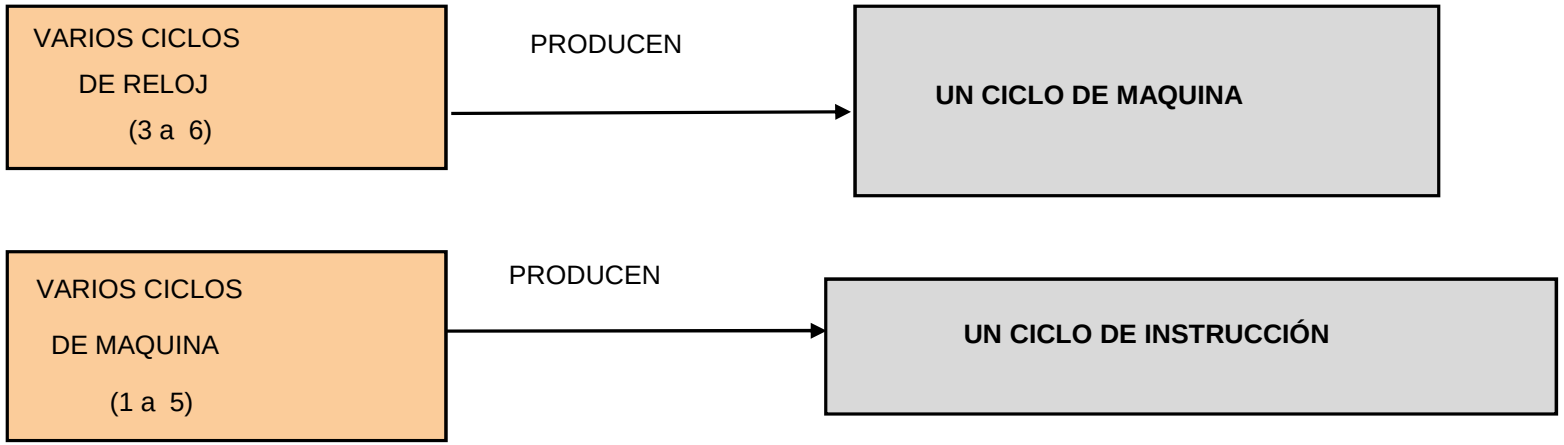


FIG. C

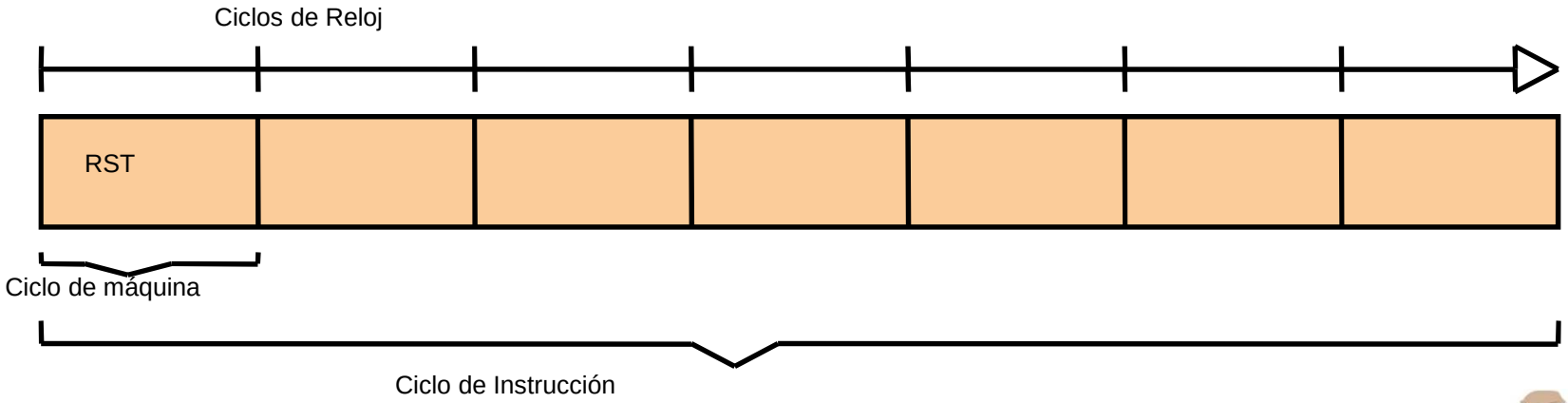


Fig. D

IRQs LINUX

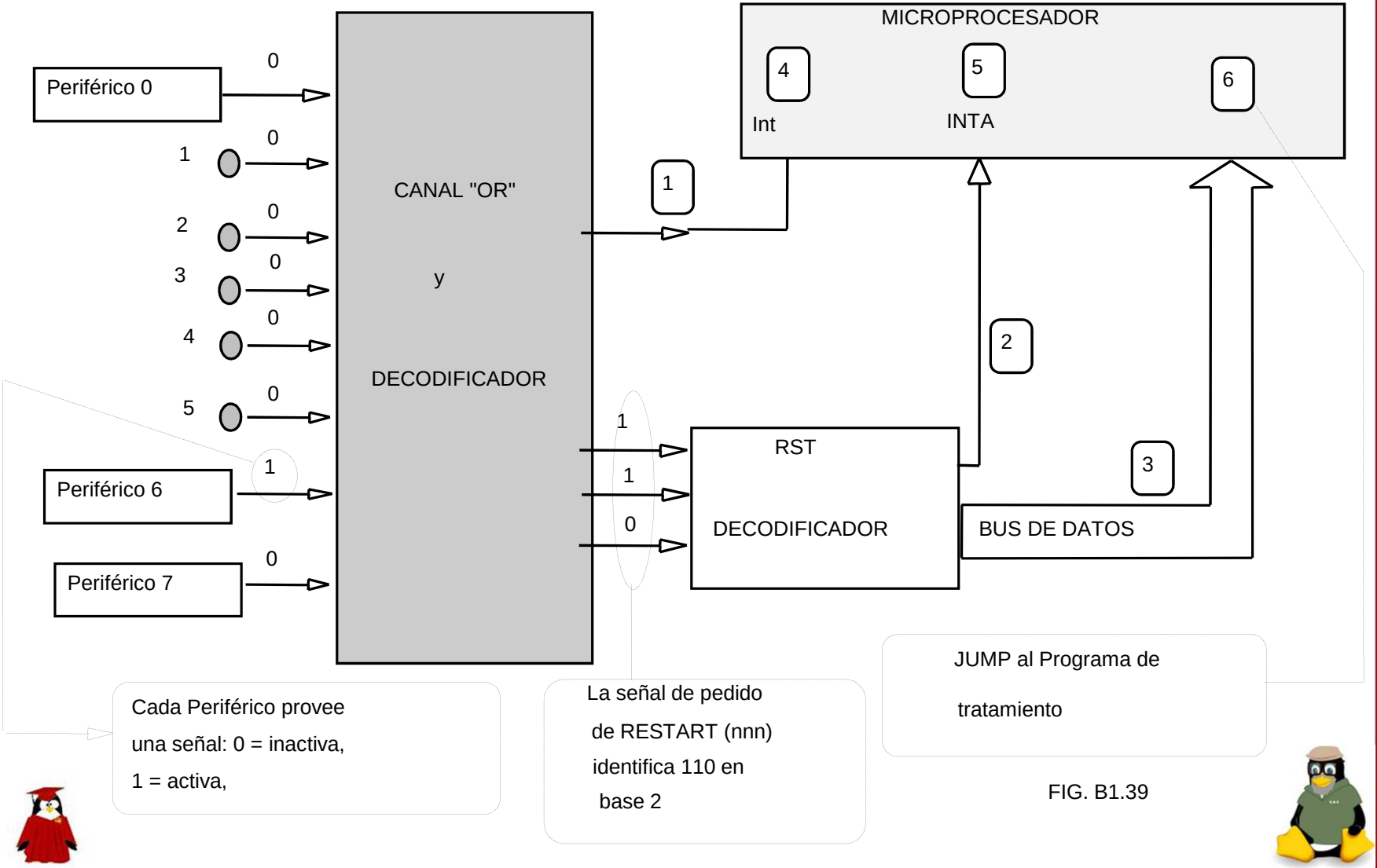


FIG. B1.39

IRQs LINUX

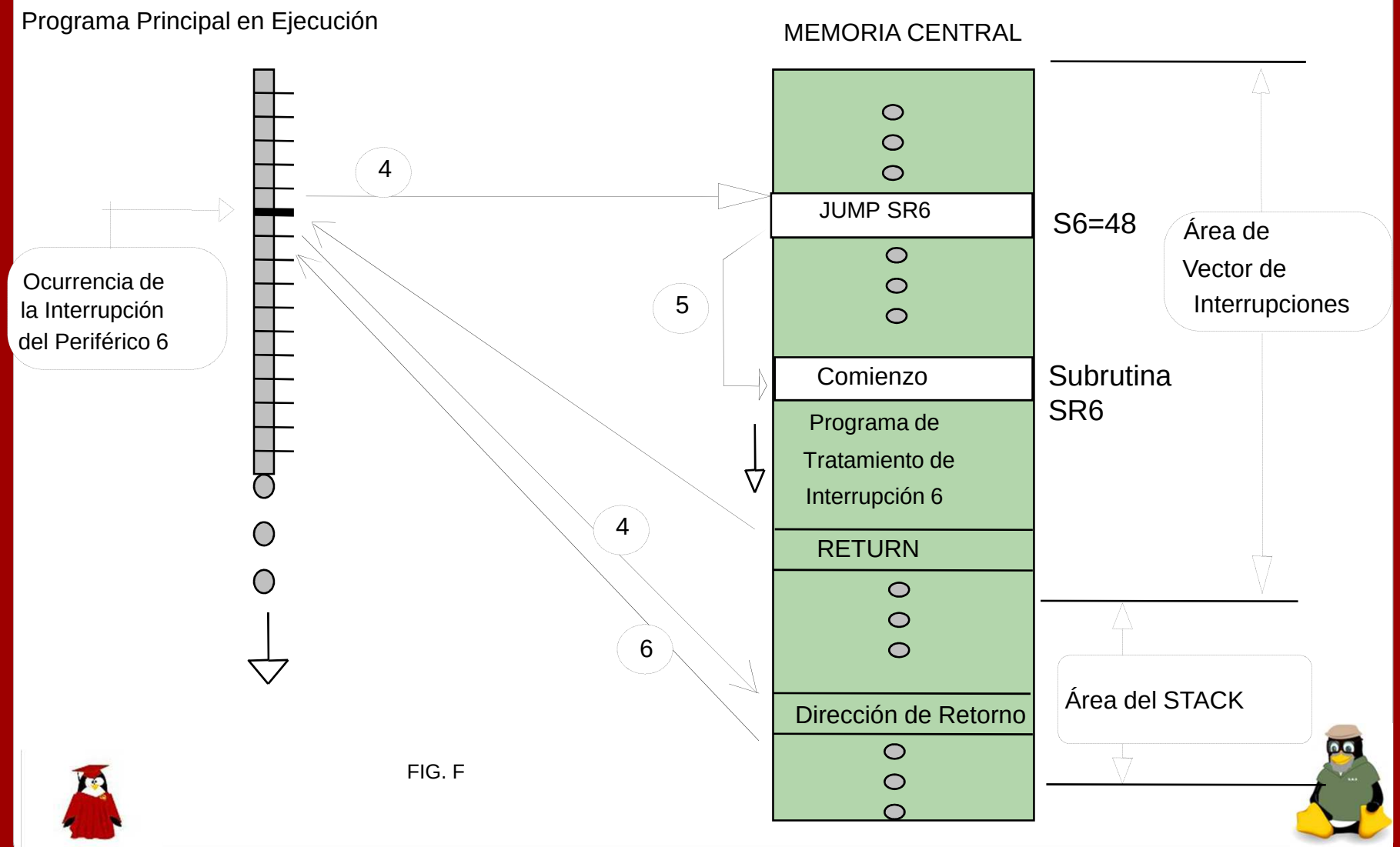


FIG. F

Bibliografia

1. Programación en Linux, con ejemplos. Kurt Wall. QUE, Prentice Hall. Madrid. 2000.
2. Sistemas Operativos. 5ta Ed. William Stalling. Pearson Prentice Hall. Madrid. 2006
3. Sistemas Operativos. 7ma Ed. William Stalling. Pearson Prentice Hall. Madrid. 2012
4. Sistemas Operativos Modernos. Andrew. S. Tanenbaum. Prentice-Hall. Interamericana S.A. Madrid, 2009.
5. Unix, Sistema V Versión 4. Rosen,Rozinsky y Farber.McGraw Hill. NY 2000.
6. Lunix, Edición especial. Jack Tackett, David Guntery Lance Brown. Ed. Prentice Hall. 1998.
7. El Libro de Linux. Syed M. Sarwar, Robert Koretsky y Syed. A. Sarwar. Ed. Addison Wesley. 2007. España.

FIN UNIDAD 5

SISTEMAS DE LLAMADAS Y SEÑALES

May the force be with you