

Universidad de Costa Rica

Facultad de Ingeniería

ECCI

Prof. Diego Villalba

BSD UNIX

Rebeca Ramírez Arroyo

B05068

2013

BSD UNIX

Inicios del BSD Unix

El sistema operativo nace después de un exitoso proceso de desarrollo del sistema UNIX. La versatilidad del código, claridad de diseño y pequeño tamaño facilitó la distribución hacia centros de investigación entre ellos la Universidad de Berkeley. El proyecto BSD (Berkeley Software Distribution) inició en 1993, se creó con el objetivo de ser multipropósito y de buen rendimiento con la ventaja de poder ser usado en varias plataformas.

Existen varias distribuciones del sistema:

- FreeBSD: se centra en aprovechamiento del uso de recursos, rendimiento del sistema y facilidad de uso.
- NetBSD: se centra en la portabilidad del sistema.
- OpenBSD: se centra en la seguridad, se han realizado búsquedas extensas de errores y fallas.
- TrustredBSD: es el más reciente y su objetivo es su certificación por el Departamento de Defensa de Estados Unidos. [1]

Se va a describir este sistema por características fundamentales que hacen de un sistema operativo diferente a los otros, donde se denotan las influencias de su origen, el público objetivo, época de elaboración y tendencia de desarrollo.

Diseño e implementación del sistema

El diseño del BSD está basado en la arquitectura de UNIX, por lo tanto lo describiremos conforme a esta última.

¿Cuáles son los principios de UNIX?

- La simplicidad de sus algoritmos, con el fin de las personas puedan desarrollar un sistema más complejo si lo desean.
- El diseño tiene un enfoque fuerte en estar amoldado para el uso de desarrollo de parte de programadores, esta característica se ejemplifica con el make y el Source Code Control System.
- Está escrito en C, que aventaja su portabilidad hardware.
- La protección, UNIX tiene un sistema de control de errores del sistema, pero en lugar de tratar de solucionar el problema ocasionado, se tiende a evitar estos problemas.[3]

Interfaz:

El sistema cuenta con programas para realizar llamadas al sistema para simplificar su uso ante el usuario, ejemplo de ello son mkdir, rmdir, cd, pwd, ls, entre otros, que son originarios del sistema UNIX.

- Uso del Shell:

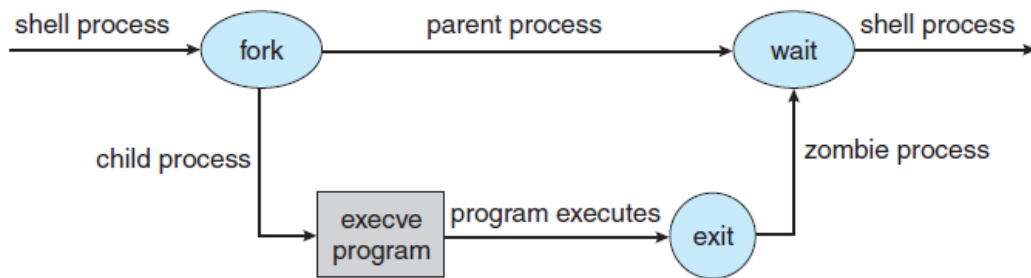
El signo % significa el acceso al C Shell, para realizar acciones existen diferentes comandos hechos para el uso del Shell tal como ls, cd, entre otros. [3]

Administración de procesos:

BSD ofrece facilidades básicas que cualquier programador agradecería, una de ellas el manejo de procesos, estos son un hilo de control en la memoria, los procesos tienen un identificador, propietario y un grupo.

Los procesos constan de un proceso padre, que es el que los inicia, que los vincula al shell, a diferencia de estos los procesos init siempre serán los primeros procesos. [2]

La imagen siguiente ilustra la ejecución de los procesos padre-hijo.



Cuando se crea un proceso el valor de retorno del proceso padre es el identificador del hijo y el del hijo es el valor 0, dicho dato es básico a la hora de programar para logra identificar la ejecución de un proceso hijo. Generalmente los procesos hijos heredan todos sus recursos del padre pero también pueden ser creados como copia creando un proceso hijo con un conjunto propio de recursos.

Los procesos suspenden su ejecución con la llamada al sistema wait, la cual retorna el id del procesos del proceso terminado y el estado de terminación. Un detalle importante es que la planificación de procesos de da con el esquema de prioridad. [4]

- Grupos de procesos:

Los grupos de procesos se crean para tener un control sobre los procesos que accesan a la terminal y para controlar y tener un mayor orden en cuanto a los signal que realiza cada proceso.

El grupo creado se va a identificar con el id del proceso que haya creado el grupo, ese grupo de va a componer de los procesos hijos de este proceso creador. Este proceso padre puede alterar en forma masiva a sus descendientes. [4]

Administración de memoria propia del BSD:

Los requisitos del BSD fueron el soporte de los espacios largos para direcciones, archivos asignados y memoria compartida. El mmap que permite la memoria compartida por varios procesos, varias versiones de BSD no la tienen, por motivos de fortalecer el soporte en redes que ofrece el sistema, pero las últimas versiones cuentan con ambos aspectos.

El cuanto a memoria virtual, su principio de diseño fue darle un uso moderado a la memoria, sin embargo, no fue diseñado para las arquitecturas multiprocesador, y en vista de su fuerte apertura en el mercado, se decidió cambiar su diseño completamente para su adaptación, este último fue basado en el sistema de memoria virtual de la Mach.

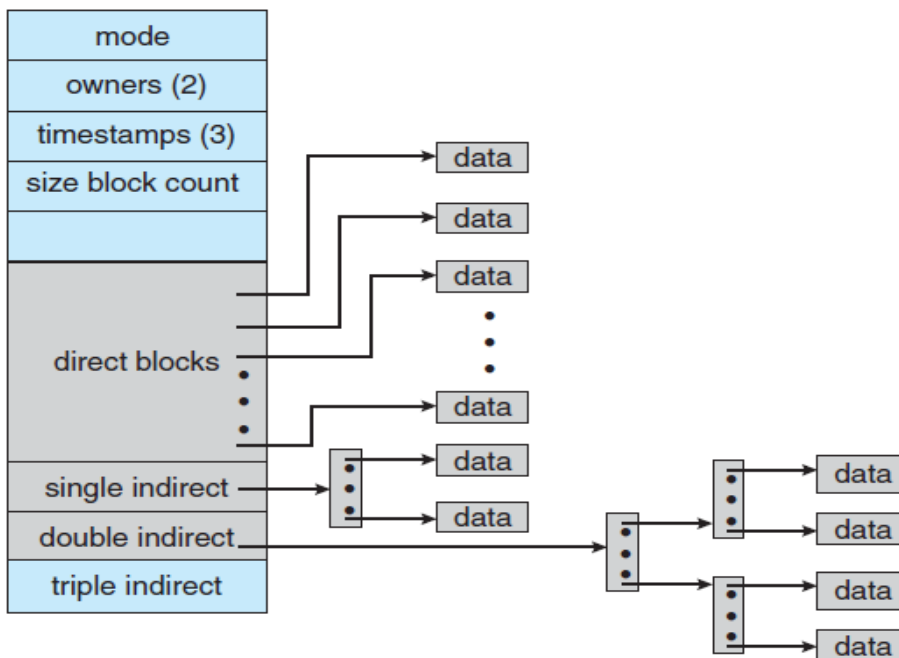
Entre la forma de administrar la memoria están:

- Los procesos pueden asignar archivos en cualquier lugar de su espacio.
- Los cambios realizados por un proceso son visibles en el espacio de direcciones de otro proceso.
- Los procesos pueden solicitar asignaciones particulares de un archivo.
- La memoria virtual copia los datos del espacio de direcciones de proceso en un búfer en el kernel.[4]

Sistema de archivos:

El sistema soporte archivos y directorios.

- Bloques y fragmentos: Un dato interesante es que los sistemas UNIX limitan el sector de disco a 1021 bytes porque el sistema contiene un gran número de pequeños archivos y para evitar una excesiva fragmentación.
- Inode: un inode es un registro de cada archivo en el disco. Este contiene un usuario, un grupo de identificadores del archivo, última fecha de modificación, entre otros detalles que son únicos de cada archivo. El inode contiene 15 punteros a disco que contienen estos datos de cada archivo. Se dividen en bloques que pueden ser "direct blocks", "indirect blocks", "single indirect blocks", "double indirect", "single indirect block pointer", "triple indirect block".



- Directorios: La referencia de un archivo desde el punto de vista del usuario es por su directorio, pero el archivo utiliza el inode para su definición, para localizar un archivo el kernel debe asignar el directorio al inode. Cuando un inode es encontrado por una llamada del sistema, la estructura del archivo es asignada a un puntero del inode.
- Estructuras del disco: Un archivo consiste de varios archivos físicos en el sistema, esto beneficia en cuanto a que si ocurre un daño en parte del disco, no todo el archivo se llega a dañar, también que se garantiza que un archivo no va a ocupar todo un bloque de memoria dejando memoria sin utilizar.

Bibliografía

1. Oscar Bonilla, "Una breve Historia de Unix". Consultado el 5/11/2013, <http://oscarbonilla.com/writing/unix-hist/>
2. "Manual de FreeBSD", Proyecto de Documentación de FreeBSD, Revisión: [43126](http://www.freebsd.org/doc/es/books/handbook/basics-processes.html). Recuperado de <http://www.freebsd.org/doc/es/books/handbook/basics-processes.html>
3. Avi Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", Ninth Edition, Apéndice "Unix BSD", 2012.
4. Marshall Kirk McKusick, Keith Bostic, Michael J. Karels, John S. Quarterman "The Design and Implementation of the 4.4BSD Operating System", 1996.