

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

Análisis de subsistemas Unix como opción de interoperabilidad de sistemas

Autor: Gabriela García Escobedo

**Tesina presentada para obtener el título de:
Ing. En Sistemas Computacionales**

**Nombre del asesor:
César Augusto Mendoza Morales**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar, organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





UNIVERSIDAD VASCO DE QUIROGA

ESCUELA DE SISTEMAS COMPUTACIONALES

**“ANALISIS DE SUBSISTEMAS UNIX COMO
OPCION DE INTEROPERABILIDAD DE
SISTEMAS”**

TESINA

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN SISTEMAS COMPUTACIONALES**

**PRESENTA:
GABRIELA GARCIA ESCOBEDO**

**ASESOR:
ING. CESAR AUGUSTO MENDOZA MORENO**

CLAVE: 16PSU0049F

MORELIA, MICHOACÁN, MÉXICO

JUNIO DE 2007

DEDICATORIAS

*A mis padres,
Con mucho amor para ustedes.*

AGRADECIMIENTOS

A Dios,

*Por siempre estar conmigo y nunca abandonarme.
Y me permite llegar hasta aquí.*

Al Ing. César Augusto Mendoza Moreno,

*Te agradezco a ti por todo tu tiempo, dedicación, esfuerzo.
Y por ser el mejor asesor que pude haber tenido..*

A Blanca Daniela,

*Con mucho amor y esperanza de llegar a ser
Un ejemplo para ti.....*

A Rafael,

Por tu apoyo para seguir adelante.

INDICE

INTRODUCCIÓN.....	- 2 -
DEFINICIÓN DEL PROBLEMA Y JUSTIFICACIÓN.....	- 3 -
OBJETIVOS GENERALES Y ESPECÍFICOS	- 4 -
CAPITULO I . MARCO TEORICO	- 5 -
1.1 Plataforma Windows	- 6 -
1.1.1 Introducción.....	- 6 -
1.1.2 Archivos	- 6 -
1.1.3 Sistemas de archivos	- 7 -
1.1.3.1 Sistemas de Archivo Orientados a disco	- 7 -
1.1.3.2 Sistemas de Archivo Orientados a Red.....	- 8 -
1.1.4 Manejo de memoria	- 9 -
1.1.5 Procesos.....	- 10 -
1.1.6 Redes	- 10 -
1.2 Plataforma UNIX/Linux	- 12 -
1.2.1 Introducción.....	- 12 -
1.2.2 Archivos	- 12 -
1.2.2.1 Sistemas de Archivos	- 14 -
1.2.2.2 Sistemas de Archivo Orientados a Red.....	- 15 -
1.2.3 Manejo de Memoria.....	- 16 -
1.2.4 Procesos.....	- 17 -
1.2.5 Redes	- 18 -
1.3 Tabla Comparativa entre Arquitecturas y Paradigmas	- 19 -
1.4 Interfaz Grafica de Windows	- 22 -
1.5 Interfaz Grafica UNIX	- 23 -
1.5.1 Antecedentes.....	- 23 -
1.5.2 Independencia Completa del Sistema Operativo.....	- 23 -
1.5.3 Portabilidad	- 23 -
1.5.4 Arquitectura Cliente-Servidor	- 23 -
1.5.5 X Window	- 24 -
1.5.6 XOrg.....	- 24 -
1.5.7 Gestor de Ventanas	- 25 -
1.5.8 Entornos de Escritorio.....	- 26 -
1.6 Diferencias entre permisos y seguridad	- 26 -
1.7 Problemáticas existentes en la interoperacion de plataformas UNIX/Linux y Win.....	- 29 -
1.7.1 Conectividad y Servicios de Redes.....	- 29 -
1.7.2 Acceso a la Información.....	- 29 -
1.7.3 Interoperatividad de datos	- 29 -
1.7.4 Administración de la Interoperatividad.....	- 30 -
1.7.5 Seguridad en la interoperatividad de ambas plataformas.....	- 30 -
CAPITULO II. PROPUESTAS DE SOLUCION	- 31 -
2.1 Introducción	- 32 -
2.2 SISTEMAS DE ARCHIVO.....	- 32 -

2.2.1 Samba.....	- 32 -
2.2.2 Compartir datos entre UNIX//Linux y Windows	- 33 -
2.2.3 NFS	- 34 -
2.3 SERVICIOS DE DIRECTORIO.....	- 36 -
2.3.1 LDAP	- 36 -
2.3.2 Active Directory	- 36 -
2.4 BASES DE DATOS	- 37 -
2.4.1 ODBC.....	- 37 -
2.5 AMBIENTES VIRTUALES.....	- 38 -
CAPITULO III. SUBSISTEMAS UNIX	- 41 -
3.1 Tecnología Microsoft Services For Unix, SFU.....	- 42 -
3.2 Tecnología CygWin.....	- 46 -
3.3 UWIN una alternativa a CygWin	- 47 -
3.4 Tabla Comparativa entre herramientas	- 51 -
CAPITULO IV . IMPLANTACION, PRUEBAS Y COMPARATIVAS	- 52 -
4.1 IMPLANTACION.....	- 53 -
4.1.1 Services for Unix	- 53 -
4.1.2 Cygwin	- 61 -
4.1.3 Uwin.....	- 67 -
4.2 PRUEBAS.....	- 68 -
4.2.1 Services for Unix	- 68 -
4.2.2 CygWin / Uwin.....	- 70 -
4.3 Comparativas entre herramientas	- 71 -
4.4 Conclusiones de Capitulo	- 71 -
CAPITULO V. CONCLUSIONES	- 72 -
—A—	
APENDICE O ANEXOS.....	- 75 -
—R—	
REFERENCIAS	- 80 -

INTRODUCCION; JUSTIFICACIÓN Y OBJETIVOS

INTRODUCCION

El paradigma de administración de los Sistemas Operativos Windows (Vista, XP, NT, 2000 o 2003) está basado en interfaces gráficas (*GUI - Graphical Users interface*); estos sistemas dedican bastantes recursos de hardware para mantener operando estas interfaces gráficas. Debido quizás a su inercia como empresa de marketing preocupada solo por la respuesta de los usuarios finales hacia su GUI, Microsoft ha creado un ambiente de trabajo colorido y musicalmente atrayente para los usuarios finales, pero salta entonces la cuestión: ¿Estas versiones de servidor de Microsoft son lo suficientemente robustas, eficientes y sobre todo, seguras en implantaciones con información importante, tareas de misión crítica, altas cargas de trabajo o necesidades de alta disponibilidad?

Mientras que el atractivo del entorno gráfico para el usuario en las estaciones de trabajo es digno sin duda de consideración, no es generalmente una cualidad deseada cuando se trata de calibrar el rendimiento y robustez de un servidor en producción. UNIX/Linux y los sistemas operativos que comparten su filosofía como GNU/Linux, FreeBSD, OpenBSD, etc., permiten entornos gráficos sin perder la posibilidad ventajosa de trabajar a través de la línea de comandos (*CLI - Command Line Interface*), por lo que un buen administrador de sistemas aprecia la libertad de evitar tales entornos gráficos que impactan el rendimiento y opta generalmente por el poderoso entorno de línea de comandos de UNIX/Linux, aunque su curva de aprendizaje sea mucho mayor.

Hacer una actualización, un parche o instalar un firewall, crear scripts de shell, la automatización a gran escala de los procesos del sistema y sus programas, asegurar la continuidad de los procedimientos, la recuperación fiable en caso de desastre, la administración remota, etcétera, es mucho más sencillo, rápido y eficaz usando línea de comandos.

Actualmente la mayoría de los departamentos de informática de las empresas tienen entornos mixtos, que incluyen tanto a Windows como a UNIX/Linux, y esta seguirá siendo la tendencia del futuro. Lograr una interoperabilidad entre sistemas resulta esencial para que se puedan construir sistemas flexibles que solucionen los problemas de hoy en día. Con dicha interoperabilidad entre sistemas se obtiene acceso a ambas plataformas mediante líneas de comando UNIX/Linux y la capacidad de usar aplicaciones basadas en UNIX/Linux en un entorno Windows.

Este proyecto analiza subsistemas UNIX, los cuales son componentes para la plataforma Windows que facilitan la interacción entre ambientes UNIX/Linux y Windows. Estos proveen servicios de plataforma para aplicaciones UNIX/Linux en Windows, esto constituye una opción para la interacción satisfactoria entre sistemas UNIX/Linux y Windows. Para analizar esta interacción se analizarán los subsistemas SFU (*Services for Unix*) de Microsoft; Cygwin de Cygnus Solutions y Uwin de AT&T, por su naturaleza podemos referenciarlos de aquí en adelante como herramientas de interoperabilidad para UNIX/Linux y Windows.

Las diferencias entre estos subsistemas UNIX/Linux reside en sus arquitecturas, licencias, madurez, alcances y limitaciones, las cuales se irán conociendo más a fondo a lo largo de este proyecto, buscando identificar sus alcances y limitaciones, su funcionalidad y probar su proceso de implantación. Se harán comparativas teóricas y prácticas buscando sus ventajas y desventajas inherentes.

DEFINICIÓN DEL PROBLEMA Y JUSTIFICACIÓN

Un sistema heterogéneo es aquel cuyos elementos cuentan con características físicas y operativas distintas entre si, pero cuentan con medios comunes que le permiten comunicarse e interactuar entre si.

Hoy en día es cada vez mas frecuente la existencia de ambientes heterogéneos, sin embargo esta interoperación entre sistemas representa un verdadero desafío para las organizaciones.

Existe una creciente necesidad de proporcionar a las empresas a través de sus centros de información, un mayor valor a su negocio; de reducir los costos, y sobre todo de consolidar rápidamente plataformas diversas. Existen administradores de UNIX/Linux que quisieran sacar un mayor provecho de Windows, y a los administradores en plataformas Microsoft, que se ven cada vez más en la necesidad de ejecutar aplicaciones UNIX/Linux cuando cuentan solo con servidores Windows.

La necesidad actual de interoperabilidad entre UNIX/Linux y Windows cada vez es mayor; hoy por hoy existen organizaciones que tienen su inversión en UNIX/Linux y esta se encuentra al final de su ciclo de vida, o están aquellos que quieren abandonar UNIX/Linux para disminuir sus costos en tecnología.

Microsoft SFU (*Services for Unix*), una implantación propietaria de subsistema UNIX, busca ayudar a resolver esas necesidades ofreciéndoles una interoperabilidad sencilla entre UNIX/Linux y Windows con una amplia colección de herramientas y utilerías estilo UNIX/Linux para administrar la plataforma Windows permitiendo a las compañías aprovechar la experiencia adquirida en UNIX/Linux por sus administradores de tecnología de la información y aplicarla a la plataforma Windows para que coexistan de manera eficaz en un entorno heterogéneo.

SFU surgió con el nombre de Interix, como un sistema completo de ejecución de aplicaciones, que permite a los clientes poner en funcionamiento de forma nativa, programas y scripts UNIX/Linux sobre sistemas operativos Windows. Interix era un producto independiente, que fue adquirido en 1999 por Microsoft y distribuido a partir de entonces de forma complementaria al sistema operativo Windows hasta el año 2002. En el 2004 se integro con el nombre de Windows Services for Unix con la versión 3.5, y desde ahí fue integrado sucesivamente, estando actualmente en su versión 6.0, incluida en las versiones de servidor de Windows Vista con el nuevo nombre de SUA o Subsistema para Aplicaciones UNIX/Linux (Subsystem for Unix-based Applications)

Existen también otros subsistemas UNIX las cuales ofrecen opciones de interoperación, como son Cygwin y Uwin; estos varían en su estructura interna y en la cantidad de software disponible. En este trabajo se analizaran estos 3 subsistemas UNIX como una opción mas de interoperatividad entre sistemas UNIX/Linux y Windows.

OBJETIVOS GENERALES Y ESPECÍFICOS

Objetivos generales

Realizar una investigación documental y experimental sobre subsistemas de aplicaciones UNIX como opción de interoperabilidad entre sistemas UNIX/Linux y Windows..

Objetivos específicos

- Explorar los subsistemas Windows Services for Unix, Cygwin, Uwin así como sus alcances y limitaciones.
- Implantación y evaluación básica de dichas subsistemas y pruebas elementales de operatividad.
- Recopilación de casos de éxito.

CAPITULO I
MARCO TEORICO

1.1 Plataforma Windows

1.1.1 Introducción

Las filosofías de diseño influyen en cientos de pequeñas decisiones de los desarrolladores. Debido a que estas filosofías son diametralmente opuestas, esas “micro decisiones” tienden a ir en direcciones opuestas y esa es lo que diferencia a los sistemas UNIX de los fabricados por Microsoft.

Windows está basado en interfaz grafica de usuario basada en el paradigma de las ventanas y de ahí su nombre en inglés. Windows y UNIX/Linux no son sistemas abiertos, pero existe entre ellos una gran diferencia y está en los estándares; en Windows son cerrados y su código fuente solo es accesible a ciertos socios comerciales tras firmar acuerdos sumamente restrictivos y de forma muy limitada a Universidades y Gobiernos; UNIX/Linux por otra parte, está basado en estándares abiertos, públicos y documentados, y aún cuando su código fuente tampoco es abierto, existen sistemas libres que siguen su misma filosofía de administración, arquitectura y funcionamiento, principalmente GNU/Linux, donde dicho código fuente es totalmente accesible para su estudio y modificación.

A continuación se abordarán diversos temas como sistemas de archivos, procesos, redes, etc., analizándolos con la intención de mostrar que ambos sistemas son diametralmente opuestos tanto en su filosofía y arquitectura; además estos aspectos son de suma importancia para poder llegar al análisis de subsistemas UNIX y así poderlo postular como opción de ínteroperación entre sistemas UNIX/Linux y Windows.

1.1.2 Archivos en Windows

Un archivo es una identidad lógica compuesta por una secuencia finita de bytes, almacenada en un sistema de archivos ubicado en la memoria de una computadora.

En Windows los archivos son agrupados en directorios dentro del sistema de archivos y son identificados por un nombre de archivo. El nombre forma la identificación única en relación a los otros archivos en el mismo directorio.

Los archivos pueden tener atributos particulares como fecha de creación, fecha última de modificación, ubicación, tipo de archivo, tamaño del archivo, hora, propietario y en las versiones mas recientes de los sistemas Windows, permisos de acceso, ejecución y escritura según el usuario que los crea o los manipula.

En algunos otros sistemas operativos los nombres de los archivos son *case sensitive*, es decir que son sensibles a las mayúsculas y minúsculas, como podría ser en el caso de UNIX; pero en Windows las mayúsculas y minúsculas no son diferenciadas a la hora de elegir el nombre del archivo.

El tamaño de un archivo esta limitado por una serie de factores, como la capacidad disponible en la memoria de la computadora y los limites impuestos por el sistema operativo o el de archivos. El tipo de un archivo es caracterizado por la organización de los datos contenidos y la interpretación que realiza el software que los escribe o los lee.

Tipos de archivos disponibles:

- Archivos de texto: son aquellos que están formados por texto, sin formato, únicamente de caracteres.
- Archivos binarios: son los archivos electrónicos que han sido guardados utilizando el código básico de las computadoras, es decir, una sucesión de ceros y unos.
- Archivos de programa: Archivo que contiene una aplicación o programa y usan las extensiones, EXE, PIF, COM, o BAT.

Es necesaria en Windows una extensión para el reconocimiento del tipo de archivo, ya que en otros sistemas operativos puede ser identificado mediante el mecanismo de los permisos. Se tiene una forma de NOMBRE.EXTENSION. Existen cientos de caracteres reservados que no pueden utilizarse en los nombres, como el asterisco por ejemplo.

1.1.3 Sistemas de archivos

Existen dos criterios para agrupar a los sistemas de archivos:

- Orientados a disco
- Orientados a la red

1.1.3.1 Sistemas de Archivo Orientados a disco

Tras dividir el disco duro en porciones llamadas particiones que hacen referencia a dispositivos en Windows identificándolos por una letra y dos puntos (*por ejemplo C:, D:, E:. etc.*), se deben formatear con un sistema de archivos.

A partir de Windows NT, Windows soporta cuatro tipos de sistemas de archivos distintos, pudiendo trabajar con los cuatro a la vez (es decir, un disco con varias particiones, en cada una un sistema de archivos distinto). Los sistemas de archivos son los siguientes:

Sistemas de Archivos	Sistemas Operativos Soportados
FAT (File Allocation Table)	DOS, Windows NT y OS/2
HPFS (High Performance File System)	OS/2 y Windows NT
CDFS (CD-ROM File System)	Windows NT
NTFS (New Technology Filesystem)	Windows NT, en adelante

A continuación se presentan a grandes rasgos, las ventajas y desventajas de los sistemas de archivo orientados a disco soportados en la plataforma Windows.

Sistema de Archivos	Ventajas	Desventajas
FAT	Poco consumo de recursos del sistema. El mejor para discos y/o particiones de menos de 200MB.	Su rendimiento decrece con particiones mayores a los 200MB. No se pueden aplicar permisos sobre archivos y directorios.
HPFS	Este es el mejor para discos y/o particiones entre los 20 y 400MB. Elimina la fragmentación almacenando en un solo bloque el archivo completo.	No es eficiente para menos de 200MB y no se pueden aplicar permisos sobre archivos y directorios.
CDFS	Sistema de archivos que permite acceder a las pistas y imágenes de los CD-ROM como si fueran archivos normales de una forma directa.	De uso exclusivo en dispositivos ópticos, nombres de archivo limitados a 32 caracteres, profundidad máxima de 8 niveles.
NTFS	Para volúmenes de más de 400MB. Recuperable (registro de transacciones). Diseñado para no ejecutarle utilerías de reparación. Es posible establecer permisos y registro de auditoría sobre archivos y directorios.	No recomendable para menos de 400MB. Consume de 1 a 5 MB de acuerdo al tamaño de la partición.

Cuando se anunció al sucesor de Windows XP, Windows Longhorn (el cual finalmente fue lanzado como Windows Vista), se decía que uno de los puntos fuertes sería el nuevo sistema de archivos WinFS¹, sustituto de NTFS. Microsoft decidió finalmente que WinFS no llegaría para Windows Vista como se había previsto, sino que WinFS será un sistema de archivos exclusivo para las plataformas SQL Server y la plataforma ADO.NET. Y por lo tanto el soporte de sistemas de archivos no se ha incrementado. Si se tiene éxito en estas plataformas sería posible que lo implementaran en el siguiente sistema operativo de Microsoft aunque quizás las cosas ya hayan cambiado para entonces.

1.1.3.2 Sistemas de Archivo Orientados a Red

El modo de Windows para compartir archivos en la red se basa en el protocolo llamado Server Message Blocks (SMB). SMB fue originalmente inventado por IBM; desde entonces diversos fabricantes (especialmente Microsoft) han ido ampliando su funcionalidad progresivamente, creando diferentes versiones de SMB. En ocasiones el cambio de versión ha conllevado a rebautizar el propio protocolo. En este sentido SMB ha recibido, entre otros, los siguientes nombres: Core Protocol, DOS Lan Manager, LAN Manager, NTLM (NT Lan Manager), y en los

¹ <http://es.wikipedia.org/wiki/WinFS>

últimos años CIFS (Common Internet File System). Todos ellos por tanto hacen referencia a SMB, aunque se diferencien en algunos detalles de su funcionalidad y/o implementación.

SMB fue popularizado por Microsoft para la comunicación de sus estaciones y servidores Windows en lo referente a servicios de archivos en impresión; generalmente se implementa sobre NetBIOS (Network Basic Input/Output System), y su principal ventaja es que al ser nativo de Windows no requiere la instalación de componentes adicionales. Adicionalmente permite compartir, impresoras, puertos seriales, etc.

SMB se basa en un esquema de tipo cliente-Servidor, donde la computadora “servidor” ofrece recursos (archivos, impresoras, etc.) que pueden ser utilizados remotamente por las computadoras “cliente” a través de la red. Así mismo es un protocolo de los denominados petición-respuesta, indicando que las comunicaciones se inician siempre desde el cliente como una petición de servicio al servidor (dicha petición se denomina precisamente SMB) que la procesa y retorna una respuesta a dicho cliente. La respuesta del servidor puede ser positiva (con resultado de procesar la petición del cliente) o negativa (mensaje de error), en función del tipo de petición, la disponibilidad del recurso, el nivel de acceso (permisos) del cliente, etc.

1.1.4 Manejo de memoria

En la administración de memoria de Windows 2000(NT 5.0) y sus sucesores Windows 2003 y Windows XP, se decidió utilizar una técnica de paginación segmentada (clustered paging) y un liberador de memoria que se activa una vez por cada segundo liberando porciones de memoria ya utilizadas o descartadas.

La gestión de memoria es la encargada de dos funciones principales. Por una parte, realizar la traducción o mapeado, del espacio de direcciones virtual a la memoria física. De forma que cuando un proceso lea en su espacio privado de direcciones, pueda acceder a la dirección correcta de memoria física. Por otro lado su otra función consiste en realizar el volcado de algunas zonas de memoria a disco, cuando ésta se encuentra sobrecargada.

Entre las partes que componen el gestor de memoria, se pueden destacar:

- Servicios del sistema para reservar, liberar, o gestionar la memoria virtual, muchos de los cuales están expuestos a través de la API Win32, o en interfaces de módulos de kernel.
- Un control para resolver excepciones a fallos de memoria.
- Algunos componentes que se ejecutan en modo sistema: liberador de memoria, un marcador de páginas de memoria modificadas, un proceso encargado de escribir las páginas marcadas a disco, etc.

Como los demás componentes ejecutados en Windows, el gestor de memoria soporta el multiproceso, es decir, permite a varios hilos de ejecución reservar recursos sin que estos interfieran entre ellos. Para cumplir este objetivo el gestor de memoria utiliza diferentes mecanismos internos de sincronización para controlar el acceso a sus estructuras de datos. Por defecto, cada proceso de usuario puede reservar más de 2GB de espacio privado de direcciones.

El gestor de memoria utiliza un algoritmo de paginación bajo demanda para saber cuando cargar páginas en memoria. Las páginas en espacio de direcciones de un proceso pueden estar libres, reservadas, o en uso. Las aplicaciones pueden reservar un espacio de direcciones para usarlo posteriormente, o pueden reservarlo y usarlo en la misma llamada de función.

1.1.5 Procesos

En la arquitectura de Windows NT y superiores, los procesos son segmentados en componentes mas pequeños llamados 'threads'. Windows soporta varias tareas al mismo tiempo. Existen dos tipos de multitarea, el apropiativo (preemptive) y el no apropiativo (no preemptive). Con la multitarea apropiativa la ejecución de un 'thread' puede ser suspendida después de un tiempo determinado por el sistema operativo para permitir que otro thread sea ejecutado. Mientras que con la multitarea no apropiativa, es el thread es el que determina cuando le regresara el control al sistema operativo para permitir que otro 'thread' sea ejecutado.

El manejador de procesos es un componente que crea, destruye procesos y tareas; este ve a los procesos como si fueran objetos; en efecto, el manejador de procesos puede ser considerado como una instancia especifica del manejador de objetos, porque dicho manejador crea, maneja, y destruye un solo tipo de objetos.

Se puede únicamente distinguir una funcionalidad adicional al manejador de objetos con la que cuenta el manejador de procesos, que consiste en el manejador de estadio de cada uno de los procesos (ejecutar, suspender, reiniciar, terminar una tarea).

Las llamadas a procedimientos locales LPC (*Local Process Call*), son usados para pasar mensajes entre dos diferentes procesos corriendo dentro de un mismo sistema Windows; estos sistemas fueron modelados utilizado como modelo las llamadas a procedimientos remotos RPC (*Remote Process Call*). Los RPC consisten en una manera estandarizada de pasar mensajes entre un cliente y un servidor a través de una red. Similarmente los LPC's pasan mensajes de un procedimiento cliente a un procedimiento servidor en un mismo sistema Windows.

Cada proceso cliente en un sistema Windows que tiene la capacidad de comunicación por medio de LPC's debe tener por lo menos un objeto de tipo puerto asignado a él; este objeto tipo puerto es el equivalente a un puerto de TCP/IP (*Transmisión Control Protocol / Internet Protocol*) en un sistema UNIX/Linux.

1.1.6 Redes

El primer tipo de implementación utilizada en un sistema de red sencillo fue denominado "Workgroups". Un "Workgroups" o grupo de trabajo es un grupo entre 20 y 200 computadoras que comparten datos y recursos (ejemplo: impresoras, scanner y otros dispositivos.) pero no requieren de un servidor central.

La falta de un servidor en este tipo de implementaciones, otorgaba una mayor sencillez a la arquitectura de un sistema ya que no se requería un Protocolo de Red. (Como TCP/IP en UNIX/Linux).

En los 80's IBM desarrollo el protocolo denominado NetBIOS (*Network Basic Input / Output System*), que solucionaba algunos de los problemas que presentaba la tecnología utilizada con anterioridad, donde la ejecución de todas las aplicaciones se realizaba exclusivamente en el MainFrame, y para distribuir esta carga fue necesario desarrollar dicho protocolo de redes (NetBIOS).

Años más tarde surgió la alianza entre IBM y Microsoft y se creó el sucesor de NetBIOS NetBEUI (*NetBIOS Extended User Interface*), y el cual empezó a ser integrado en plataformas Windows 3.1 bajo el nombre de "Windows for Workgroups", y posteriormente en Windows 95 y 98.

A pesar de los beneficios que trajo NetBEUI (el subdividir el sistema en varios "Workgroups" y tener un servidor central) éste seguía teniendo un problema, debido a la sencillez de su diseño este protocolo no podía ser ruteado—debido a que su estructura no es jerárquica, tiene una estructura denominada "flat-address space", esto es, utiliza nombres para ubicar los "Hosts" (ventas1,ventas2,ingeniería4), esto es suficiente para una LAN (*Local Área Network*) pequeña , sin embargo, cuando la información (paquetes) requiere ser enviada por una LAN grande o WAN (*Wide Área Network*) es necesario utilizar direcciones lógicas (TCP/IP).

1.2 Plataforma UNIX/Linux

1.2.1 Introducción

UNIX/Linux se orienta en dirección contraria a la tendencia de hacer invisible al usuario el sistema operativo; permite el uso de todas sus bibliotecas, llamadas al sistema, y herramientas internas, aunque su uso requiere un alto nivel de especialización; es posible adaptarlo a las necesidades de una empresa o grupo de trabajo, reduciendo el uso de recursos y aumentando la rapidez.

Este tipo de sistemas ha encontrado un gran apoyo en varios ámbitos: en el ámbito universitario; en los laboratorios de informática de muchas empresas que buscan versatilidad para el desarrollo de aplicaciones en plataformas heterogéneas, etc.

Por otra parte la administración en las diferentes versiones de servidores UNIX/Linux puede hacerse vía entornos gráficos *GUI (Graphical User Interface)* o a nivel de *CLI (Command Line Interface)*. Comparando a UNIX/Linux con la administración basada en GUI de los servidores de Windows, estos últimos son incapaces de usar guiones de automatización y procedimientos de la forma que lo hacen todas las versiones de UNIX y GNU/Linux. La automatización administrativa es posible gracias a la línea de comandos y el scripting de UNIX/Linux.

UNIX/Linux ha estado históricamente asociado a una cierta dificultad de uso y administración. Se puede decir que, a cambio, se obtiene un enorme control sobre el sistema, que puede crecer a la medida de nuestras necesidades.

1.2.2 Archivos

En UNIX/Linux se accede a las particiones mediante entradas específicas en la rama /dev del árbol jerárquico de archivos, y por este motivo en UNIX/Linux no existe el concepto de unidades (como en Windows) ya que todo está accesible bajo el directorio principal. Por ejemplo, en UNIX/Linux no se accede a la primera unidad de discos flexibles mediante la orden *A:* como en Windows, sino que hay que “montarla”.

De este modo tenemos dos conceptos nuevos:

Montar: Es agregarle ramas a el árbol invertido del Sistema de Archivos de UNIX/Linux para que utilice un dispositivo dado.

Desmontar: Es el quitarle una rama al árbol invertido del Sistema de Archivos de UNIX/Linux para que deje de utilizar el dispositivo asociado.

Para los sistemas UNIX mas populares se tienen los sistemas de archivo siguientes:

Sistema Operativo	Descripción
*BSD	UFS (Unix File System) FFS (Fast File System)
OPENSOLARIS	ZFS (Zettabyte File System)
HP-UX	JFS (Journaling Filesystem)

Linux además dispone de una gran cantidad de sistemas de archivos adicionales entre los cuales se encuentran EFS, ReiserFS, EXT2, EXT3, XFS, HFS, etc. ya que el soporte es muy amplio.

En UNIX/Linux antes de extraer un dispositivo hay que “desmontarlo”, para que de esta manera se almacenen correctamente los datos pendientes a grabarse en dicho dispositivo. Un ejemplo muy claro de ello, es que la puerta de un lector de CD-ROM, que haya sido “montado”, simple y sencillamente no podrá ser abierta físicamente hasta que se “desmante”.

Para UNIX/ Linux todo dispositivo de sistema es un archivo y en general todo el sistema se accede mediante descriptores de archivos.

Tipos de Archivos:

- Comunes (*Ordinary Files*): textos, programas fuente, objeto, ejecutables, etc. En este sistema operativo los archivos son solo una secuencia de bytes⁷ (no tiene en cuenta registros, ni índices, ni reconoce marca de eof). La asignación de espacio es dinámica.
- Especiales (*Special File*): dispositivos físicos (terminales, impresoras cintas, etc.) y otros (por ejemplo: named pipes).
- Directorios (*Directory Files*) son archivos que a su vez contienen referencias de archivos. Los directorios se crean para mantener la información en forma ordenada, agrupan archivos con alguna relación. A un directorio contenido dentro de otro directorio se le llama subdirectorio y el directorio que lo contiene se le llama directorio padre.
- Enlaces simbólicos

Otra característica de UNIX/Linux es que se distinguen entre mayúsculas y minúsculas, tanto en sus archivos como sus comandos. En UNIX/Linux los nombres de archivo no tienen extensión por lo que un archivo puede contener tantos puntos como queramos en su nombre; de hecho el nombre puede tener hasta 256 caracteres de longitud. Aquí también existen caracteres que son reservados y el asterisco vuelve a ser uno de ellos. Los archivos que son ejecutables se distinguen en este caso por algo que no se ve en el nombre sino en el permiso de ejecución.

El comando `ls` de UNIX/Linux muestra las características de un archivo de la siguiente manera:

```
-rwxr-xr-x 15 gabyta dba 4096 nov 2 18:43 pruebas/
```

Aquí se indica prácticamente todo lo que hace falta saber sobre un archivo con respecto a la seguridad y a los permisos, empezaremos a analizar desde la izquierda y veamos que significa y su diferencia con Windows.

El primer guión (-) indica que ese archivo NO es un directorio (o carpeta) ya que si lo fuera aparecería una “d” en su lugar. UNIX/Linux trata a los directorios meramente como otro archivo, aunque de tipo especial, y los permisos tienen un significado ligeramente diferente cuando hacen referencia a un directorio y a un archivo.

Los tres caracteres siguientes, corresponden a los permisos del propietario, que pueden ser distinto de su creador original, ya que UNIX/Linux, permite a los usuarios “dar” archivos a otros

usuarios. La “r” indica que el propietario tiene derecho a leer (read) el archivo, la “w” significa que puede escribir (write) en él, y la “x” permite al propietario del archivo ejecutarlo, si fuera un programa. En UNIX/Linux, no hay ninguna asociación entre la extensión y la ejecutabilidad del archivo, de hecho, la mayor parte de los archivos UNIX/Linux no tienen ninguna extensión. Lo único que determina que el archivo es un ejecutable, es su permiso. Por lo tanto, el archivo debe recibir el permiso de ejecución para poder ejecutarlo.

Los caracteres quinto, sexto y séptimo corresponden a los permisos de los miembros del mismo grupo del propietario del archivo. Finalmente los últimos tres caracteres del primer grupo corresponden a los permisos del resto del mundo (al resto de usuarios). Tal y como vemos, UNIX/Linux únicamente posee tres permisos básicos que son: lectura escritura y ejecución.

Siguiendo el ejemplo, el siguiente carácter, que es un número “15” indica que hay 15 vínculos duros contra el archivo. Un “vínculo duro” concede otro nombre al mismo archivo. Sigue siendo un único archivo real guardado en el disco duro, pero significa que hay 15 entradas de directorio que apuntan a ese archivo. No hay límites prácticos al número de vínculos duros que puede haber con un mismo archivo.

Los dos grupos siguientes del nombre, son el “propietario” en este ejemplo el propietario es “gabyta,” y el grupo de archivo “dba”. Aunque suelen ser los nombres del usuario y grupo, también podrían ser un número si el usuario y grupo del archivo no tuvieran una cuenta en el sistema.

A continuación aparecen el tamaño del archivo, a fecha y la hora en que se creó o se modificó y el nombre del archivo “pruebas”. Podemos observar que no hay nada significativo en cuanto al nombre que aparezca primero, todos se tratan por igual, y el borrado de un vínculo no borra el archivo, solo esa referencia al mismo.

1.2.2.1 Sistema de Archivo

UNIX/Linux se organiza como un árbol invertido donde cada nodo es un archivo (ya sea un directorio, un archivo común, un dispositivo, etc.) Esto no es una simplificación, el sistema representa como archivos a los dispositivos (disco rígido, CDS, floppys, algunas conexiones de red, etc.) que están conectados. La estructura del sistema de archivos es jerárquica, donde el punto más alto de la jerarquía es la raíz del sistema de archivos, denotada por una barra (slash) “/”.

Las primeras versiones de UNIX utilizaban un sistema de archivos al que se referían simplemente como FS. FS funcionaba bien para los pequeños discos de aquel tiempo pero la tecnología avanzó y los discos comenzaron a crecer. Fue cuando BSD optimizó esto en el FFS (*Fast File System*).

UFS es el acrónimo para denominar al sistema de archivos UNIX file System (UFS) el cual es un sistema de archivos utilizado por varios sistemas operativos UNIX y tipo-UNIX. Es un derivado de Berkeley Fast File System (FFS), el cual fue desarrollado en los laboratorios Bell y permite nombres de archivo de hasta 250 caracteres.

Cabe resaltar que UNIX además de toda la gran cantidad de sistemas de archivo que soporta también puede acceder a los sistemas de archivo de Windows respetando las limitaciones

inherentes (es incapaz de fijar características a los archivos según el estilo UNIX si Windows no lo soporta). Este soporte es limitado además, por el alcance logrado en su estudio, el cual se realiza a través de ingeniería inversa, debido a que se trata de un sistema cerrado que no está documentado públicamente.

1.2.2.2 Sistemas de Archivo Orientados a Red

Los sistemas UNIX/Linux por su parte, utilizan tradicionalmente el sistema de archivos de red NFS (*Network File System*), que fue desarrollado originalmente por Sun Microsystems para compartir los sistemas de archivos en la red. También existe Samba que es una implementación libre del protocolo SMB con las extensiones de Microsoft, y funciona sobre los sistemas operativos GNU/Linux y otros UNIX.

Los usuarios del sistema NFS pueden acceder a archivos que se encuentran físicamente en máquinas remotas de una forma transparente, como si se tratara de archivos locales.

Los beneficios que NFS proporciona son entre otros:

- Las estaciones de trabajo locales utilizan menos espacio de disco debido a que los datos se encuentran centralizados en un único lugar, pero pueden ser accedidos y modificados por varios usuarios, de tal forma que no es necesario la réplica de información.
- Los usuarios no necesitan disponer de un directorio “home” en cada una de las máquinas de la organización. Los directorios “home” pueden crearse en el servidor NFS para posteriormente poder acceder a ellos desde cualquier máquina a través de la infraestructura de la red.
- También se pueden compartir a través de la red dispositivos de almacenamiento como disquetes, CD-ROM y unidades zip. Esto puede reducir la inversión en dichos dispositivos y mejorar el aprovechamiento del hardware existente en la organización.

El sistema NFS está dividido en dos partes el servidor y uno o más clientes. Los clientes acceden de forma remota a los datos que se encuentran almacenados en el servidor.

El protocolo NFS tiene múltiples usos prácticos. Algunos se enumeran a continuación:

- Compartición de la unidad de CD-ROM entre varias máquinas. Esto resulta ser más barato y una forma más conveniente para instalar software en varias máquinas.
- En grandes redes puede ser más adecuado configurar un servidor central de NFS en el cual se almacenan todos los “homes” de los distintos usuarios. Estos directorios se pueden exportar a través de la red de tal forma que los usuarios pueden trabajar con el mismo directorio independientemente de la máquina que utilicen.

1.2.3 Manejo de Memoria

UNIX/Linux utiliza dos técnicas de memoria: swapping y memoria virtual. Lo estándar en UNIX/Linux es un intercambio de segmentos de un proceso entre memoria principal y memoria secundaria, llamado swapping lo que significa que se debe mover la imagen de un proceso al disco si este excede la capacidad de la memoria principal, y copiar el proceso completo a memoria secundaria. Es decir, durante su ejecución, los procesos son cambiados de y hacia memoria secundaria conforme se requiera.

Si un proceso necesita crecer, pide más memoria al sistema operativo y se le otorga una nueva sección lo suficientemente grande para acomodarlo. Entonces, se copia el contenido de la sección usada al área nueva, se libera la sección antigua y se actualizan las tablas de descriptores de procesos. Si no hay suficiente memoria en el momento de la expansión, el proceso se bloquea temporalmente y se le asigna espacio en memoria secundaria. Se copia a disco y posteriormente, cuando se tiene el espacio adecuado (lo cual normalmente ocurre en algunos segundos) se devuelve a memoria principal.

Esta claro que el proceso que se encarga de los intercambios entre memoria y disco (llamado swapper) debe ser especial y jamás podrá perder su posición privilegiada en memoria central. El kernel se encarga de que nadie intente siquiera interrumpir este proceso del cual dependen todos los demás. Este es el proceso 0 cuando se decide traer a la memoria principal un proceso en estado “listo para ejecutar”, se le asigna memoria y se copian allí sus segmentos. Entonces el proceso cargado compete por el procesador con todos los demás procesos cargados. Si no hay suficiente memoria, el proceso de intercambio examina la tabla de procesos para determinar cual puede ser interrumpido y llevado al disco.

Surge la pregunta, ¿cual de los posibles procesos que están cargados será desactivado y cambiado a memoria secundaria? Los procesos que se eligen primero son aquellos que están esperando operaciones lentas de E/S, o que llevan cierto tiempo sin haberse movido a disco. La idea es tratar de repartir en forma equitativa las oportunidades de ejecución entre todos los procesos, tomando en cuenta sus historias recientes y sus patrones de ejecución.

Otra pregunta es ¿Cuál de todos los procesos que están en disco será traído a memoria principal? La decisión se toma sobre la base del tiempo de residencia en memoria secundaria. El proceso más antiguo es el que se llama primero, con una pequeña penalización para los grandes.

Cuando UNIX/Linux opera en maquinas mas grandes, suele disponer de manejo de memoria de paginación por demanda. En algunos sistemas el tamaño de la pagina es de 512 bytes; en otros 1024; para reemplazo se utiliza un algoritmo que mantiene en memoria las paginas empleadas mas recientemente LRU (*Least Recently Used*) que en español significa algo así como menos utilizado recientemente o frecuentemente, y entonces aquí el liberador de memoria se utiliza solo cuando es necesario y no a intervalos regulares como en Windows donde se ejecuta cada segundo).

Un sistema de paginación por demanda ofrece muchas ventajas en cuanto a flexibilidad y agilidad en la atención concurrente de múltiples procesos y proporciona además memoria virtual, es decir la capacidad de trabajar con procesos mayores que el de memoria central. Estos esquemas son bastante complejos y requieren el apoyo de hardware especializado.

1.2.4 Procesos

El manejo de procesos en UNIX/Linux es por prioridad. En algunas versiones se maneja también un ajuste dinámico de la prioridad de acuerdo al tiempo que los procesos han esperado y al tiempo que ya han usado el CPU. El sistema provee facilidades para contabilizar el uso de CPU por proceso y una pila común para todos los procesos cuando necesitan estar ejecutando en modo privilegiado (cuando hicieron una llamada al sistema). UNIX/Linux permite que un proceso haga una copia de sí mismo por medio de la llamada 'fork', lo cual es muy útil cuando se realizan trabajos paralelos o concurrentes; también se proveen facilidades para el envío de mensajes entre procesos. Recientemente Sun Microsystems, AT&T, IBM, Hewlett Packard y otros fabricantes de computadoras llegaron a un acuerdo para usar un paquete llamado ToolTalk para crear aplicaciones que usen un mismo método de intercambio de mensajes.

La ejecución de un proceso en UNIX/Linux se divide en dos niveles: nivel de usuario y nivel kernel. Cuando se produce una llamada al sistema se pasa del modo usuario al modo kernel. Este analiza la llamada, la ejecuta y devuelve el control a modo usuario. Esta diferenciación de modo se produce porque los procesos en modo usuario pueden acceder a sus instrucciones y datos, pero no a instrucciones y datos del kernel o de otros usuarios; mientras que el modo kernel puede acceder a todos los datos e instrucciones del sistema. Hay instrucciones privilegiadas a las que solo se puede acceder en modo kernel, el cual reside permanentemente en memoria.

Existen varios mecanismos por medio de los cuales los procesos se pueden comunicar. En primer lugar, es posible crear canales unidireccionales para la comunicación entre dos procesos, de forma que uno escribe una secuencia de bytes que es leída por otro. Estos canales se llaman pipes (tuberías). El uso normal de los pipes es la comunicación entre un proceso padre y un proceso hijo, de forma que uno escribe y el otro lee. Los procesos se pueden sincronizar mediante pipes, ya que cuando un proceso intenta leer de un pipe vacío, permanece bloqueado hasta que hay información disponible.

Para que los programas puedan acceder a los dispositivos físicos UNIX/Linux los integra dentro del sistema de archivos mediante los archivos especiales, de forma que cada dispositivo se le asigna un archivo que se encuentra normalmente en el subdirectorio /dev.

Los archivos especiales son accedidos de la misma forma que los ordinarios. Es decir que los puede abrir, leer y escribir. De esta manera no es necesario ningún mecanismo adicional para realizar la entrada/salida. Otra de las ventajas que posee es que las reglas de protección de archivos se aplican a los dispositivos, lo que permite controlar los accesos.

Los archivos especiales se dividen en dos categorías: dispositivos de bloques y dispositivos de caracteres. Los dispositivos de bloques consisten en una secuencia numerada de bloques, de forma que cada bloque puede ser accedido de forma individual. Se usan para discos. Los dispositivos de caracteres se usan normalmente para dispositivos que leen o escriben secuencias de bytes, tales como terminales, impresoras, mouse, etc.

1.2.5 Redes

Los principales avances de la comunicación de red en los sistemas UNIX, surgieron en 1983, con la aparición del sistema 4.2 BSD (*Berkley Software Distribution*) que integraba TCP/IP (*Transmission Control Protocol / Internet Protocol*) y el API (*Application Programming Interface*) de Sockets. A partir de este código surgen BSD Networking Release 1.0 en 1989, también denominado NET-1, hasta el surgimiento de 4.4 BSDLite (NET-3) en 1994 en el que precisamente está basado el *nix FreeBSD.

Lo importante de los sistemas *nix, es que desde sus primeros pasos en Redes, todas sus implementaciones nativas están basadas en el protocolo TCP/IP. UNIX/Linux dispone de dos principales protocolos de red: TCP/IP y UUCP.

El protocolo TCP/IP tiene sus orígenes en un proyecto de investigación fundado en Estados Unidos por el DARPA (*Defense Advanced Research Projects Agency*), Agencia de Proyectos Avanzados de Investigación en Defensa) en 1969. Esta fue una red experimental, la red ARPANET, que pasó a ser operativa en 1975, después de haber demostrado ser un éxito.

En 1983, fue adoptado como estándar el nuevo conjunto de protocolos TCP/IP, y todos los nodos de la red pasaron a utilizarlo. Cuando ARPANET por fin dio paso a Internet (con la propia ARPANET integrándose en su existencia en 1990), el uso del TCP/IP se había extendido a redes más allá de la propia Internet. Las más destacables son las redes locales UNIX.

Otra aplicación muy importante en redes TCP/IP es NFS, ya que es otra forma de hacer transparente la red, porque básicamente permite montar jerarquías de directorios de otras máquinas, de modo que aparezcan como sistemas de archivos locales. Por ejemplo, todos los directorios "home", o personales, de los usuarios pueden estar en una máquina servidor central, desde la cual montan los directorios el resto de máquinas de la LAN. El efecto de esto es que los usuarios pueden acceder a cualquier máquina, y encontrarse a sí mismos en el mismo directorio.

UUCP es una abreviatura de UNIX-to-UNIX Copy (Copia de UNIX a UNIX). El cual es un viejo mecanismo usado para transferir archivos, correo electrónico y noticias entre máquinas UNIX. Comenzó siendo un paquete de programas para transferir archivos sobre líneas serie, programar esas transferencias e iniciar la ejecución de programas en el lugar remoto. Ha experimentado grandes cambios desde su primera implementación a finales de los setenta, su principal aplicación es todavía en redes de área metropolitana WAN (*Wide Area Network*, red de área amplia) basadas en enlaces telefónicos. Actualmente, UUCP ya no está confinado al entorno UNIX. Hay versiones comerciales disponibles para diversas plataformas, incluyendo AmigaOS, DOS, TOS de Atari, etc.

Una de las principales desventajas de las redes UUCP es su bajo ancho de banda. Por un lado, el equipo telefónico establece un límite rígido en la tasa máxima de transferencia. Por otro lado, los enlaces UUCP raramente son conexiones permanentes; en su lugar, los nodos se llaman entre sí a intervalos regulares.

A pesar de estas limitaciones, aun hay muchas redes UUCP funcionando en todo el mundo, utilizado principalmente por aficionados, ya que ofrecen acceso de red a usuarios privados a precios razonables. La razón fundamental de la popularidad del UUCP es que es baratísimo

comparado con tener un equipo conectado al Internet. Para hacer un nodo UUCP, todo lo que se necesita es un módem, software UUCP, y otro nodo UUCP que suministre correo y noticias.

La idea que hay detrás de UUCP es bastante simple: como su nombre indica, básicamente copia archivos de un nodo a otro, pero también permite realizar ciertas acciones en el nodo remoto

1.3 Tabla Comparativa entre Arquitecturas y Paradigmas

Características	Windows	UNIX/Linux	Punto medio
Sistemas con Estándares Públicos y documentados	No	Si	UNIX/Linux cuenta con estándares públicos y bien documentados, es por esto que toda persona que quiera hacer uso de estos estándares esta en plena libertad de hacerlo, y por el contrario en Windows esto no es posible, y el acceso a esta información y al código fuente es siempre con condiciones muy restrictivas además de solo ser accesibles a gobierno o instituciones académicas; es por esto que su escasa documentación y poco soporte a estándares lo vuelve una plataforma no muy viable para el desarrollo a futuro.
Sistema de Archivos orientados a Disco	FAT16, FAT32, VFAT, NTFS	FFS, UFS, ReiserFS, XFS, JFS, ext2, ext3, etc...	UNIX/Linux soporta por su arquitectura mas modular una gran cantidad de sistemas de archivos. En cuanto al control de acceso y ejecución actualmente ambas plataformas están equilibradas porque ambos utilizan filosofías tradicionales sobre permisos y derechos de archivos, al menos desde Windows NT en adelante. En ambos casos, existen nuevos desarrollos con paradigmas nuevos para la organización y control de los sistemas de archivos.
Manejo de memoria	Windows utiliza una	UNIX, utiliza un algoritmo de	Al hacerse en Windows a intervalos de un segundo, esta

	técnica de paginación segmentada (clustered paging) y un liberador de memoria que se activa una vez por segundo	paginación adaptable LRU (Last Recently Used, menos utilizado frecuentemente) y el liberador de memoria que se ejecuta solo cuando es necesario.	propiedad es utilizada por los virus para provocar la famosa “saturación”, de la memoria (desbordamiento de pila) que bloquea los CPU y obliga a resetear, incluso los servidores Otra ventaja de UNIX/Linux sobre Windows es que UNIX/Linux no utiliza memoria si el programa no esta abierto o corriendo, en Windows, aun estando cerrado el programa, puede estar utilizando porciones de memoria, para la eventualidad de que se vuelva a abrir y acelerar este proceso.
Mecanismos de Procesos	Manejador de Procesos	La ejecución de un proceso se divide en: nivel de usuario y	En Windows utiliza un manejador de procesos el cual es un componente que crea, maneja y destruye procesos y tareas. En UNIX/Linux la ejecución de un proceso se divide en dos niveles: nivel de usuario y nivel de kernel, es decir cuando se produce una llamada al sistema se pasa del modo usuario al modo kernel este analiza la llamada, la ejecuta y devuelve el control al modo usuario. Existe esta diferenciación de modo porque los procesos en modo usuario pueden acceder a sus instrucciones y datos pero no a las instrucciones y datos del kernel o de otros usuarios; mientras que el modo kernel puede acceder a todos los datos e instrucciones del sistema. Ya que hay instrucciones privilegiadas a las que solo se puede acceder en modo kernel y el cual reside permanentemente en memoria.
Redes	NetBIOS, NetBEUI	TCP/IP	Los protocolos que Windows utiliza como NetBIOS y NetBEUI son ideales para una LAN pequeña, sin embargo, cuando la información

			(paquetes) requiere ser enviada por una LAN grande o WAN es necesario utilizar u un protocolo de comunicación como TCP/IP. Además de que UNIX/Linux usa el protocolo TCP/IP de forma nativa.
Sistemas de archivos orientados a redes	SMB	NFS	<p>Todos los servidores SMB tienen otra gran ventaja respecto de las soluciones NFS, ya que estas suelen ser significativamente más rápidos en las transferencias de archivos, especialmente al tratar archivos grandes.</p> <p>Pero también existen grandes ventajas por parte de NFS el cual evita la replica de información y de esta manera se utiliza menos espacio en disco, y, además se cuenta con un acceso transparente como si se tratara de archivos locales.</p> <p>También se pueden compartir a través de la red dispositivos y esto reduce la inversión y a la vez se obtiene un mejor aprovechamiento del hardware existente. Sin embargo la compartición de NFS es primitiva, pudiendo limitar solamente su acceso por dirección IP, 3 tipos de acceso (lectura, escritura y control total) además de que no es posible asignar directivas de compartición basado en usuarios y grupos.</p>

Tabla 1. Tabla Comparativa UNIX/Linux

1.4 Interfaz Grafica de Windows

En el caso de Windows no existe la capa de separación (la cual existe en UNIX/Linux). Actualmente el kernel de Windows es altamente dependiente de la capa grafica, así que no puede prescindir de ella. En Windows no existe la posibilidad de siquiera decir *quiero usar las librerías graficas de Windows, pero quiero usar el entorno de ventanas de Mac OS X* o algo por el estilo, Ambas capas están totalmente ligadas y no se puede despegar una cosa de la otra.

Esto le resta estabilidad al sistema, ya que por si algún motivo se tiene un fallo o simplemente se ‘cuelga’, la interfaz grafica, el sistema en general se ve afectado y debido a que no es posible reiniciar solo la interfaz grafica (como en sistema UNIX/Linux), hay que reiniciar todo el sistema operativo con la consecuente pérdida de disponibilidad del sistema en general.

En el nuevo sistema operativo de Microsoft, Windows Vista, se incorpora una nueva interfaz llamada ‘Aero’. Además se dispone de una API denominada *Windows Presentation Foundation* (conocida como Avalon en entornos de desarrollo) dedicada a la creación de ventanas para el sistema utilizando para ello gráficos vectoriales con la ayuda de XAML (*Extensible Avalon Markup Language*: Lenguaje de Formato para Extensibilidad de Avalon, un lenguaje basado en XML utilizado para la creación de interfaces gráficas de usuario en Windows Vista) y DirectX Avalon, que es un subsistema de presentación de GUI basado en vectores.

Es importante mencionar que una de las novedades que plantea el nuevo lenguaje XAML es una separación de la interfaz grafica de la aplicación de código, esto le permite a las personas usar diversas utilidades graficas para crear el layout y el diseño de la parte grafica de la interfaz de usuario sin que afecten la parte lógica del programa. Todos estos cambios, reflejados apenas en la nueva versión del sistema operativo (lo cual implica que están aún poco difundidos ya que la mayoría de sistemas siguen siendo versiones anteriores), no se traduce en una diferencia con respecto a la cerrada dependencia entre la capa gráfica y el sistema operativo, aunque si un avance en cuanto a la posible modularización del entorno gráfico y presunta separación del kernel de Windows.

1.5 Interfaz Gráfica UNIX

1.5.1 Antecedentes

En los primeros años de las computadoras, la comunicación entre el software y los usuarios se realizaba enteramente a modo de comandos, en una Terminal o incluso algunos métodos mucho más complicados como las tarjetas perforadas. Sin embargo a mediados de los años setenta, los ingenieros del centro de investigaciones de Xerox crearon un nuevo y revolucionario sistema de interacción entre el usuario y la máquina. Se trata de la interfaz gráfica de usuario (GUI), el cual se basa en la utilización de 'ventanas' e 'iconos' para la representación de aplicaciones y un 'ratón' y un 'puntero' para manejarlas. Gracias a esto, el manejo de las computadoras se hizo increíblemente mucho más fácil e intuitivo.

En los ochenta, las GUI's empezaron a llegar a usuarios reales de la mano de Apple Computer con su computadora personal 'Lisa' y posteriormente Macintosh. Rápidamente los desarrolladores de UNIX se dieron cuenta de que necesitaban un GUI para su sistema operativo. No obstante, debido a que los sistemas UNIX por lo general corrían en computadoras de alto rendimiento, servidores y poderosas estaciones de trabajo, aquel sistema debía ir mucho más allá de las GUI's planeadas para computadoras personales como las de la Apple² o Microsoft, y de esta manera fue así como surgió el sistema X Window o simplemente X.

1.5.2 Independencia Completa del Sistema Operativo

A diferencia de los sistemas de GUI de Apple o Microsoft, X fue diseñado para estar aislado completamente del núcleo (*kernel*) del sistema operativo, lo que resulta en una mejor estabilidad, ya que si por algún motivo X tiene un fallo o simplemente 'se cuelga', el sistema en general no se ve afectado. También resulta en una mayor versatilidad, debido a que es posible reiniciar X, sin reiniciar el sistema operativo, en incluso instalar diferentes versiones e implementaciones en un mismo núcleo

1.5.3 Portabilidad

X se puede portar fácilmente a cualquier plataforma que soporte ANSI y POSIX, debido a esto es posible encontrarlo en diferentes sistemas operativos y arquitecturas de hardware, desde una poderosa estación gráfica de Silicón Graphics hasta una PC; gracias a esta característica de X fue posible la creación de implementaciones para PC basados en procesadores x86 (*xfree86*, *XOrg*) e incluso en sistemas operativos como Microsoft Windows (*cygwin/x*) y MacOSX (*X11 for mac OSX*).

1.5.4 Arquitectura Cliente-Servidor

UNIX siempre ha estado orientado a redes. El sistema X Window no podía pasar desapercibido y es por eso que X está diseñado para poder trabajar en redes mediante un sistema de cliente-servidor. De esta manera, un servidor X se encarga de manejar el hardware necesario para una GUI: la tarjeta de video, el ratón y el teclado; y brindar una interfaz para poder trabajar sobre

² <http://usuarios.lycos.es/hv1102/desarrollo.html>

estos dispositivos. Un cliente X en cambio es cualquier aplicación grafica la cual debe conectarse a un servidor X para poder mostrarse en la pantalla. Para esto, la aplicación cliente hace uso de ‘*Xlib*’ que es el conjunto de librerías de programación que contienen lo necesario para tal labor. La intercomunicación entre el cliente y el servidor de X se realiza gracias a un protocolo de red llamado *protocolo X*, el cual puede trabajar sobre otros de bajo nivel como TCP/IP.

Por lo general en un sistema de escritorio GNU/Linux, tanto el cliente como el servidor de X se ejecutan en la misma maquina, y el proceso de comunicación entre los dos se realiza de forma completamente transparente. Sin embargo, también es posible ejecutar el cliente y el servidor en equipos diferentes con lo cual se obtiene una enorme flexibilidad. Por ejemplo es posible utilizar remotamente, en una computadora cliente, un programa con GUI que se este ejecutando en otra computadora (servidor). Incluso ambas computadoras pueden tener hardware y sistemas operativos distintos debido a que el cliente no necesita saber nada de lo que maneja el servidor.

1.5.5 X Window

El sistema X Window comenzó su desarrollo en el Instituto Tecnológico de Massachussets (*MIT*, por sus siglas en ingles), con el apoyo de IBM. X es básicamente un conjunto de programas, librerías, protocolos, y controladores que conforman todo lo necesario para poder usar GUI’s en UNIX. Fue diseñado para ser flexible y robusto. Algunas de las características más importantes de X son su independencia del sistema operativo, su portabilidad, su diseño en forma de cliente-servidor y por supuesto el hecho de ser software libre.

Desde la versión 11 de X (*X11*), desarrollada en el MIT, este sistema ha estado cubierto bajo una licencia de software libre compatible con la Licencia Publica General de GNU. Gracias a esto el sistema X Window se convirtió rápidamente en un estándar en cuestión de interfaces graficas en sistemas UNIX, ya que toda la industria podía participar activamente en el proyecto. Sin embargo la licencia de X11 no tiene copyleft, lo que significa que los trabajos derivados pueden ser distribuidos bajo otros términos. Esto ha causado que muchas empresas desarrolladoras de UNIX creen versiones propietarias de X. No obstante, el corazón de X todavía sigue siendo uno de los proyectos mas importantes de software libre.

1.5.6 XOrg

Xfree86 es una implementación del sistema X Window System, fue escrito originalmente para sistemas operativos UNIX funcionando en ordenadores compatibles IBM PC. En la actualidad esta disponible para muchos sistemas y plataformas.

Xfree86 es open source y software libre publicado bajo licencia Xfree86.1.1.

El proyecto Xfree86 es desarrollado por el Xfree86 Project Inc.; siendo su desarrollador David Dawes.

Xfree86 provee una interfaz grafica cliente-servidor entre el hardware (ratón, teclado y sistemas gráficos) y un entorno de escritorio que provee un sistema de ventanas así como una interfaz estandarizada de aplicación API.

Xfree86 es independiente de la plataforma, extensible y puede utilizarse en la red.

Un cambio de licencia en febrero del 2004 a partir de la versión 4.4.0 provocó la bifurcación X.Org apoyada por empresas y desarrolladores descontentos con presuntas incompatibilidades con la popular licencia GPL³ (*General Public License* o Licencia Pública General). Esto ha provocado una caída en la popularidad de Xfree86, siendo reemplazado por X.Org en algunas distribuciones de GNU/Linux y en algunos sistemas BSD.

Los sistemas X.Org y Xfree86 son muy similares, de hecho el sistema base de X.Org es Xfree86. La principal diferencia entre ellos es la licencia aplicada a los paquetes. Muchas distribuciones han decidido usar el paquete X.Org, pero varias usan a un Xfree86.

X.Org es una implementación de código abierto del sistema X Window System, que surge como bifurcación del proyecto Xfree86.

Algunas de las características de X.Org son las siguientes:

- Ventanas translucidas
- Ventanas con animación al minimizarse
- Ventanas con sombra
- Soporte para escritorios como Looking Glass de Sun

En los años 80's, la introducción del Apple Macintosh hizo advertir a todos la necesidad que había de interfaces gráficas en las computadoras de escritorio. Más o menos al mismo tiempo Microsoft empezó con el marketing para su sistema operativo orientado a GUI, Ambos Windows y Macintosh fallaron al no separar las tareas del sistema operativo de la interfaz gráfica, ya que integraron inseparablemente ambas cosas. Sin embargo, Apple al apostar en los 90's a UNIX, es decir al rediseñar desde cero su sistema operativo basado en un port de UNIX BSD logró separar en OSX su interfaz gráfica de las tareas del sistema operativo.

1.5.7 Gestor de Ventanas

La filosofía de diseño de X es parecida a la de UNIX, 'herramienta no política'. Esto significa que X no impone una tarea como debe hacerse. En lugar de eso proporciona herramientas al usuario y es responsabilidad suya saber que hacer con ellas.

Esta filosofía se extiende a X, ya que no impone como debe verse las pantallas, como moverlas con el ratón, que teclas usarse para moverse entre ventanas (por ejemplo ALT+TAB en Microsoft Windows), como deben ser las barras de título en cada ventana, si tienen o no botón de cerrar, etc.

En lugar de eso delega esta responsabilidad a una aplicación llamada 'gestores de ventanas'. Hay docenas de gestores de ventanas disponibles para X: Alter Step, BlackBox, ctwm, Enlightenment, sawfish, twm, window maker, etc. Cada uno de estos gestores de ventanas tienen un aspecto diferente algunos soportan 'escritorios virtuales', otros permiten personalizar las teclas para manejar el escritorio; otros tienen un botón de 'inicio' o similar otros tienen 'diseños modificables', permitiendo un cambio completo del aspecto mediante un nuevo diseño.

³ <http://www.gnu.org/licenses/gpl.html>

1.5.8 Entornos de Escritorio

Los propósitos de entorno de escritorio es aportar marcos de trabajo adicionales a las aplicaciones para que su apariencia y comportamiento sean visualmente consistentes. Para ilustrar estas diferencias, clasificamos algunos de los proyectos más conocidos en la categoría a la que pertenecen.

- Visualizador de gráficos: X11, Xorg
- Gestor de ventanas: sawfish, metabox
- Escritorio: Gnome, kde

X11 es el sistema grafico estándar de facto en el mundo UNIX. Acompaña a Linux, a los *BSD y a muchos sabores comerciales de UNIX. Los entornos de escritorio CDE, KDE, y GNOME se ejecutan sobre el.

1.6 Diferencias entre permisos y seguridad

Windows usa niveles de privilegios, es decir que cada cuenta de usuario determina las acciones que el usuario puede realizar en la red, y tiene tres niveles de configuración los cuales son: administrador, invitado, usuario y la cual solo una es asignada a cada cuenta de usuario.

- Privilegio de administrador: El cual tiene derechos administrativos completos en la red.
- Privilegio de invitado: Es uno de los tres niveles de privilegios que pide asignársele a la cuenta de un usuario, esta cuenta generalmente se utiliza para iniciar una sesión, la cual debe de tener el privilegio de invitado.
- Privilegio de usuario: Las cuentas con privilegio de usuario son usuarios normales de la red, probablemente la mayoría de las cuentas de la red tienen el privilegio de usuario.

Cada usuario se puede encontrar en un solo nivel de configuración y este solo posee los permisos y restricciones de dicho nivel al cual pertenezca.

Windows tiene un enfoque de permisos similar al de UNIX/Linux pero con la diferencia de que UNIX /Linux cuenta con un súper usuario llamado “root” el cual tiene acceso a todos los permisos sobre el sistema.

Los permisos de cualquier archivo siempre le permitirán la lectura, escritura y ejecución – esto significa que el súper usuario puede hacer cualquier cosa en el sistema, nada le será negado--.

En este caso UNIX/Linux se encuentra en desventaja frente a Windows ya que aunque utilizan un enfoque similar en UNIX/Linux, existen tareas que nadie, ni siquiera un administrador pueden realizar si no es el súper usuario.

El paradigma de seguridad que Windows utiliza parte de su actitud inherente de seguridad que es la “*permisividad*”, que significa que la mayoría o todas las rutinas se consideran “*confiables*” a menos que se configure lo contrario, el resultado es un débil modelo de seguridad de Windows, ya que la aparente facilidad con la que el código malévolo puede dañar los recursos del sistema operativo de Microsoft, como se puede ver en la frecuencia de asaltos de virus en todo el mundo.

En Windows debido a su gran cantidad de fallos que se encuentran día a día (inclusive muchos de ellos no llegamos ni a conocer) surgen los llamados “parches” o “actualizaciones” disponibles a través de Internet, aunque muchas otras con elevados precios en las obligadas actualizaciones de sistemas completos. Además Windows requiere (debido a estas actualizaciones) la presencia física e interacción directa para su aplicación y si esto lo multiplicamos por un numero grande de estaciones resulta sumamente impractico y muy costoso.

Reiniciar el sistema en Windows es una actividad ya considerada rutinaria para solucionar problemas, es una técnica muy común en la administración de sistemas Microsoft, a pesar de ser muy costosa y antiproduktiva en verdaderas grandes estaciones de trabajo.

En Windows cuando existe una vulnerabilidad en el programa que se ejecuta como administrador, un ataque exitoso puede permitir que el acceso se haga de los derechos y personalidad completa con la que se ejecutaba el programa y si este es un administrador las consecuencias pueden llegar a ser muy serias.

La política de UNIX/Linux en cuanto a seguridad se refiere por defecto es la “denegación” hasta que los puntos de acceso se configuren de manera adecuada. Puesto que el sistema UNIX/Linux esta diseñado para soportar múltiples usuarios dispone de muchos modos para que los usuarios accedan al sistema y muchas herramientas para comunicación entre usuarios y entre maquinas diferentes. Sin embargo, en el mundo actual hay motivaciones suficientes para que personas no autorizadas irrumpen en sistemas ajenos, desde la simple emoción hasta el daño malicioso o el robo comercial de datos y programas; es por esto que es necesario equilibrar la facilidad de acceso para los <<amigos>> con la prevención del acceso para los <<enemigos>>.

La seguridad en un sistema UNIX/Linux tradicional esta basada en la separación entre el espacio del kernel (*kernel space*) y espacio del usuario (*user space*) y en los tradicionales permisos Read, write, execute para propietario, grupo y todos los demás.

El código que se ejecuta en el espacio del kernel no tiene restricción alguna en el sistema: es decir tiene acceso directo a todo el hardware, puede utilizar cualquier operación sin acudir a las capas de abstracción que el kernel ofrece (y de hecho es su función – por ejemplo: los módulos que permiten al kernel entender diferentes sistemas de archivos y ofrecer a los programas de usuario un modelo abstracto uniforme se ejecutan en el espacio del kernel). El código que se ejecuta en espacio del kernel, además, tiene acceso a toda la memoria de la computadora, es capaz de manipular la tabla de procesos e inclusive la de modificar datos de los diferentes procesos.

En la mayoría de los sistemas UNIX/Linux, hay únicamente un proceso corriendo en espacio del kernel, precisamente en el kernel, de varias implementaciones de UNIX/Linux, tienen una implantación de *microkernel*, lo que significa que el kernel incluye únicamente una fracción de código privilegiado, y las demás facilidades de bajo nivel (por ejemplo manejo de memoria, manejo de sistemas de archivos, manejo de red) son implementadas por procesos privilegiados adicionales. Esto nos da una mayor separación funcional, lo que lleva a una mayor limpieza en el código, pero al mismo tiempo lleva a una implementación más compleja.

Los programas que utiliza el usuario final, tales como terminales u otras aplicaciones con ventanas, residen en el espacio del usuario. Como es lógico estas aplicaciones necesitan

interaccionar con el hardware del sistema, pero no lo hace directamente, sino a través de las funciones que soporta el kernel.

Todos los programas a excepción del kernel y –en su caso- estos programas privilegiados adicionales trabajan en espacio de usuario. Esto significa que tienen ciertas restricciones.

En UNIX/Linux se tiene definido un sistema simple y efectivo de permisos para cada objeto existente en nuestro árbol de directorio. Al solicitar una operación sobre él, el kernel verifica si el usuario bajo el cual está corriendo dicho proceso, o alguno de los grupos a los que pertenece, tiene derecho de llevar a cabo la operación indicada sobre el objeto. Para esto, cada objeto tiene definido un usuario dueño y un grupo, y además permisos de lectura, escritura y ejecución. Este simple esquema típicamente lo vemos representado como los tres juegos de: `rwx`, permiso de lectura, escritura y ejecución para usuario, grupo y resto del mundo.

Existe una problemática con el esquema tradicional de UNIX/Linux, ya que si bien para muchas aplicaciones el esquema de UNIX/Linux es adecuado, dista mucho de ser perfecto. El tener un usuario capaz de absolutamente todo. El tener a un administrador con capacidad de leer, eliminar o modificar cualquier dato en el sistema hace que un sistema UNIX/Linux no sea suficiente para ciertas aplicaciones donde hacen falta garantías de confidencialidad e integridad – en muchos casos gubernamentales o militares-.

En UNIX/Linux existe la posibilidad de ejecutar programas, incluso con atribuciones de “root”; sin embargo es de alto riesgo que tras una saturación de buffer o denegación de servicio puedan ejecutarse códigos como administrador. Una aplicación con defectos en su programación pone en riesgo al sistema en la medida de los privilegios del usuario que la ejecute o por una mala configuración, como podría ser un programa antivirus ejecutado como root, ya que si este está mal programado, a través del uso del fallo, un usuario con malas intenciones podría acceder, modificar o borrar archivos de sistema debido a la naturaleza de un software antivirus. O un software bien diseñado pero mal configurado, puede poner en riesgo, por ejemplo, `nfs` que se ejecuta como root, si está mal configurado (por ejemplo un carácter de espacio en un lugar inadecuado en el archivo de configuración *exports*), puede lograr que usando `nfs` los archivos del sistema puedan ser borrados o accedidos.

Con los años ha habido un desplazamiento definido en la filosofía hacia una mayor seguridad y la versión Unix System V Release 4, o SVR4 puede llegar a ser bastante segura. Un administrador de sistema experimentado puede controlar totalmente el acceso al sistema, y el sistema UNIX/Linux es ahora tan seguro o incluso más como la mayoría de los sistemas operativos. Las versiones de SVR4 han sido certificadas en los niveles de seguridad B2 y B3 del Departamento de Defensa de los Estados Unidos. Sin embargo, las cuestiones de seguridad son complejas, debido a la existencia de tantos subsistemas, todo debe estar correctamente ajustado para alcanzar una seguridad óptima. En distribuciones GNU/Linux modernas se ha avanzado sensiblemente en este rubro con la adición de módulos al kernel específicamente, SELinux (*Security-Enhanced Linux*), que contribuye con la implantación de control de acceso obligatorio (*Mandatory Access control*) permitiéndole restringir no solo en base a propietarios u operaciones, sino también contextos de acceso.

Como podemos ver en sus diferentes aspectos, tanto Windows como UNIX/Linux tienen ventajas y desventajas, y es por eso que surge la necesidad de encontrar una forma para obtener las ventajas de UNIX/Linux, con las facilidades que nos ofrece Windows; el lograr una inter

operación nos proporcionara un paradigma distinto, uno complementario, el cual nos permitirá obtener las ventajas de cada sistema operativo pero minimizando sus propias desventajas con las que cuenta cada plataforma.

1.7 Problemáticas existentes en la interoperación de plataformas UNIX/Linux y Windows.

En entornos informáticos de hoy en día es necesaria la interoperación entre sistemas. Los datos, las aplicaciones, los niveles de administración, las herramientas y las tecnologías de la red permiten que estos sistemas puedan funcionar conjuntamente.

El objetivo es el de encontrar soluciones interoperables que nos permitan conseguir objetivos como el de reducir costos totales y maximizar el valor empresarial. Pero en el camino de diseñar e implantar estas soluciones aparecen algunas problemáticas al momento de integrar ambas plataformas; en este subtema abordaremos algunas de ellas.

1.7.1 Conectividad y Servicios de Redes

El primer punto de interoperación implica tener equipos que puedan hablar entre si a través de una red. Esto significa que los equipos deben usar el mismo protocolo y que debe existir alguna manera de identificar los recursos de red.

Para conseguir este nivel mínimo de interoperación entre equipos de Windows y estaciones UNIX/Linux, la conectividad entre los entornos debe ser confiable. La elección obvia es usar TCP/IP, ya que Windows proporciona compatibilidad integrada con este protocolo, que es el estándar de red utilizado en entornos UNIX/Linux.

Una ventaja que ofrece la infraestructura común TCP/IP es la compatibilidad con servicios FTP, HTTP y Telnet; a través de los servicios FTP y HTTP los usuarios pueden copiar archivos a través de redes de sistemas heterogéneos y, a continuación, manipularlos localmente como archivos de texto o incluso como documentos de Word.

1.7.2 Acceso a la Información

El siguiente nivel de interoperación consiste en permitir el acceso a la información dentro de un entorno heterogéneo. Esto significa interoperatividad entre sistemas, lo que permite a los usuarios obtener acceso a recursos compartidos de archivos e impresoras independientemente del entorno del servidor.

1.7.3 Interoperatividad de datos

En entornos computacionales mixtos, los datos esenciales pueden estar guardados en una combinación difícil de manejar entre bases de datos y sistemas de almacenamiento. Es necesario contar con estándares y tecnologías que incluyan incluso servicios de transformación de datos que les den a los usuarios la capacidad de acceder y consultar la información que necesitan sin tener que preocuparse de donde resida o como esta almacenada.

1.7.4 Administración de la Interoperatividad

También se presenta un problema para los administradores que trabajan en entornos mixtos de sistemas, ya que enfrentan un número significativo de aspectos administrativos, como dar mantenimiento a usuarios y claves de acceso a múltiples sistemas, proporcionar autenticación segura y encontrar medios para monitorear rendimiento y sucesos eficientemente, siendo importante encontrar herramientas que nos simplifiquen las tareas administrativas del sistema, en sistemas múltiples.

1.7.5 Seguridad en la interoperatividad de ambas plataformas

Es necesario tomar en cuenta a la hora de interoperar dos plataformas distintas, la seguridad. Se debe prestar especial atención en este punto para lograr una buena interoperación de sistemas sin que nos ocasione comprometer datos y sistemas por un sistema en el entorno mixto mal protegido.

Suelen presentarse problemas como:

- La sincronización de contraseñas.
- Acceso a recursos compartidos.
- Comprobación de permisos de acceso de los usuarios.
- Permisos de archivos



CAPITULO II
PROPUESTAS DE SOLUCION

2.1 Introducción

Los entornos heterogéneos hoy en día son cada vez más frecuentes dentro de las organizaciones; y dado a esto la interoperabilidad entre sistemas se ha convertido en una característica esencial.

Un sistema heterogéneo es aquel cuyos elementos cuentan con características físicas y operativas distintas entre sí, pero cuentan con medios comunes que le permiten comunicarse e interactuar entre sí. La necesidad de la heterogeneidad, surge por diversas razones:

- Algunos sistemas son mejores para ciertas aplicaciones (compiladores gráficos, capacidad de procesamiento)
- Algunos sistemas de cómputo proveen servicios que no están disponibles para otros sistemas.
- Los recursos económicos para adquirir determinados sistemas o equipos de cómputo.

Aunado a esto que las organizaciones buscan reducir sus costos de informática y sacar el máximo partido de sus inversiones logrando la máxima eficiencia; todas estas necesidades de una organización no siempre se pueden llegar a cumplir con una sola solución. Es por esto que surge la necesidad de crear entornos heterogéneos.

El principal objetivo de la interoperación es lograr un completo y transparente funcionamiento entre plataformas. La mayoría de los usuarios no saben ni necesitan saber si sus archivos residen en un sistema de archivos local o en una red de UNIX/Linux o de Microsoft Windows. Tan solo les preocupa poder acceder a ellos; o si su correo está protegido por un sistema antispam que no podría ser ejecutado en su actual sistema operativo; o una protección antivirus más competitiva que las disponibles en su entorno; o usan un servidor de mensajería de plataforma diferente, etc. El acceso transparente a los usuarios es la clave del éxito en un entorno heterogéneo. Mismo que de lograrse adecuadamente se obtendrá un mayor rendimiento en el desempeño de su trabajo, una disminución de tiempos, optimización y una mayor calidad para la empresa.

Hoy en día es cada vez más frecuente la existencia de dichos ambientes heterogéneos, sin embargo esta interoperación de sistemas representa un verdadero desafío para las organizaciones, ya que se exige un alto grado de conocimiento por parte del personal, trabajo y en ocasiones implica elevados costos; es por esto que es recomendable hacer un estudio de factibilidad para saber si es redituable para la empresa realizar esa inversión.

2.2 SISTEMAS DE ARCHIVO

2.2.1 Samba

El primer problema en acceder archivos de una plataforma Windows es el formato o sistema de archivos (FileSystem) en el que están basados.

SMB (Server Message Block), es parte del protocolo NetBEUI desarrollado por Microsoft e IBM que permite la comunicación entre discos e impresoras en sistemas Windows. SMB opera de la forma petición-respuesta, donde los clientes envían sus peticiones, las cuales están contenidas en bloques de mensajes de servidor (*smb*) al servidor, este los recibe, los interpreta y envía de regreso la respuesta al cliente. Cuando una computadora comparte un recurso a través

de la red por medio de smb se convierte en un servidor en este escenario. Cuando una computadora accede a un recurso compartido, se convierte en un cliente.

En UNIX/Linux se logra esta funcionalidad con un proyecto de código abierto basado en ingeniería inversa del protocolo original y que es denominado SAMBA⁴ que permite que un servidor (*Host*) UNIX/Linux, pueda acceder recursos en plataformas Windows (95,98, NT, 2000), al igual que permite a estas plataformas (Windows) acceder recursos en servidores UNIX

SAMBA es un conjunto de comandos y servicios UNIX que habla el protocolo SMB. Mediante el soporte de este protocolo, Samba permite a los servidores UNIX comunicarse con el mismo protocolo de red que los protocolos de Microsoft Windows. De este modo, una maquina UNIX con Samba puede enmascararse como servidor en la red Microsoft y ofrecer los siguientes servicios:

- Compartir uno o mas sistemas de archivos
- Compartir impresoras, instaladas tanto en el servidor como en los clientes.
- Ayudar a los clientes, con visualizador de clientes de red.
- Autenticar clientes comparándose contra un dominio de Windows
- Proporcionar o asistir con un servidor de resolución de nombres WINS.
- Servidor de tiempo de Windows.
- Sustituir un Controlador primario de dominio
- Integrarse a Active Directory de manera casi total.

2.2.2 Compartir datos entre UNIX/Linux y Windows

Si se ejecuta el protocolo SMB en servidores y estaciones de trabajo UNIX/Linux, en el sistema no sera necesario ejecutar el protocolo completo sino que mediante el uso de comandos se podra tener acceso a recursos Windows. No, obstante, lo mas frecuente es utilizar clientes basados en Windows para tener acceso a archivos almacenados en servidores UNIX/Linux. Esto supone un gran reto ya que este tipo de redes son las que mas comúnmente se utilizan en las organizaciones.

Debido al excelente rendimiento de la entrada/salida de datos en los sistemas operativos UNIX/Linux, el rendimiento de un servidor Samba es equivalente al de un servidor Microsoft tanto en velocidad como en fiabilidad.

De hecho las pruebas comparativas realizadas por IT Week Lab⁵ en Mayo del 2003, muestran que la versión disponible de Samba supera el rendimiento de Windows 2003 Server.

⁴ <http://us4.samba.org/samba/>

⁵ <http://www.itweek.co.uk/itweek/news/2085218/samba-extends-lead-win-2003>
<http://www.kegel.com/nt-linux-benchmarks.html>

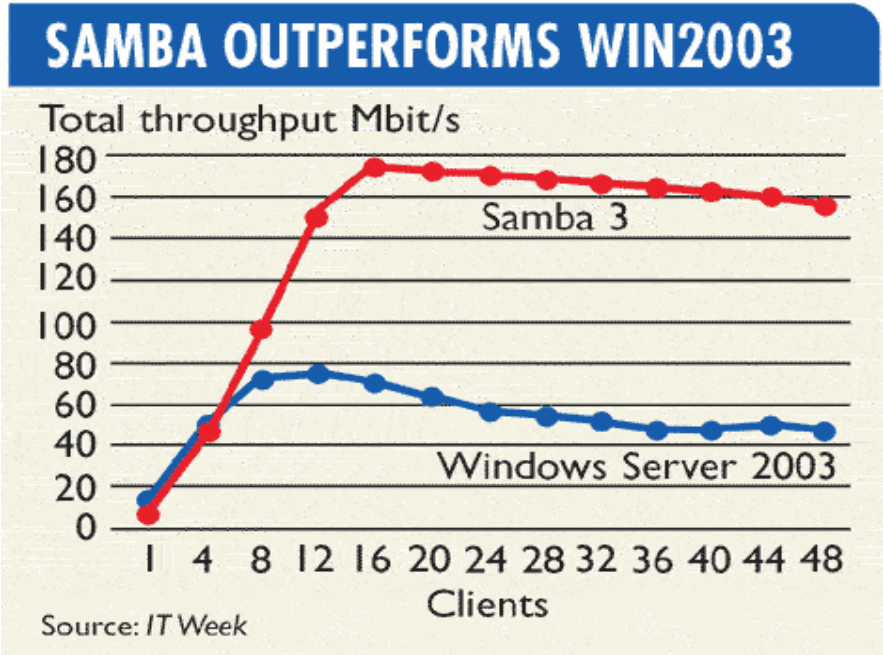


Fig. 1 Tabla comparativa Samba vs. Win Server 2003

Para algunas empresas el rendimiento de un servidor Samba para UNIX/Linux cubre las necesidades de impresión y archivos de un grupo de clientes Windows y es una alternativa muy atractiva a la costosa y menos fiable opción de un servidor Windows. Para organizaciones a gran escala, las principales plataformas UNIX como Sun Solaris y HP-UX corren una versión especialmente elaborada de Samba dentro de sus núcleos para dar soporte a SAN (Red de área de almacenamiento) y/o NAS (red adjunta de almacenamiento) a grandes grupos de clientes Windows. Un servidor UNIX/Linux puede dar servicio simultáneamente a clientes Windows vía Samba, y a clientes UNIX/Linux vía NFS.

2.2.3 NFS

El NFS o Network File System, permite que una maquina UNIX/Linux conectada a una red pueda montar un sistema de archivos de otra maquina e interactuar sobre el como si fuera propio. De esta manera, constituye un medio de compartición de archivos totalmente transparente para el usuario de la maquina cliente.

NFS no es en realidad un sistema de archivos físico, sino una capa de abstracción del sistema de archivos real (ext2, UFS, FFS, etc.) que permite el montaje de este remotamente.

Por ejemplo, si nuestro servidor es una maquina UNIX/Linux que exporta por NFS un directorio llamado, pongamos, /usr/export cuyo sistema de archivos es ext3, y tenemos uno o varios clientes Windows que quiere montar de manera remota ese directorio como alguna unidad, F:,G:,H: etc, no será necesario que nuestro Windows tenga soporte para ext3, sino que simplemente lo tenga para ser cliente NFS, y el cual viene incluido en SFU.

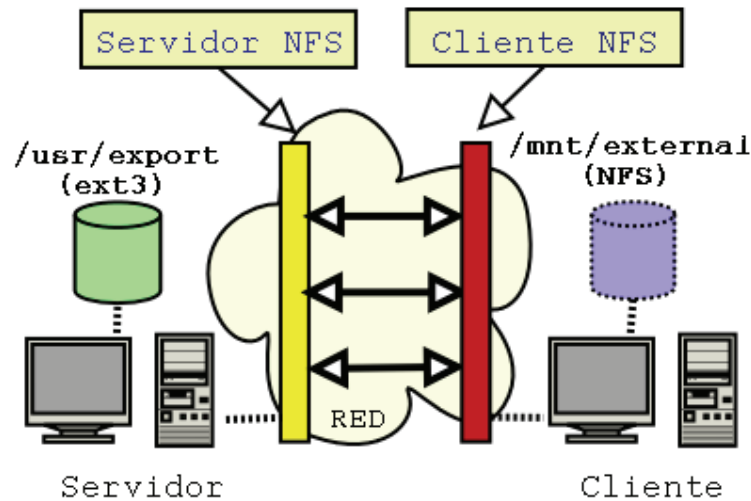


Fig. 2 Esquema NFS

Todos los principales sistemas operativos de UNIX/Linux, integran características de uso compartido de archivos NFS.

En una red mayoritariamente UNIX/Linux donde las máquinas clientes que estén basadas en plataforma Windows, estas se enfrentan al reto de poder acceder a sistemas NFS o a particiones nativas de UNIX/Linux. Esto lo pueden llevar a cabo instalando el NFS para clientes en su entorno operativo de Windows y de esta manera la máquina cliente podrá tener acceso a los archivos e impresoras que los servidores UNIX/Linux manejan.

En Windows se presenta de dos maneras: cliente NFS, que permite que los clientes Windows tengan acceso a archivos de servidores UNIX/Linux y el servidor de NFS para Windows NT Server que permiten que las estaciones de trabajo y los servidores de UNIX/Linux tengan acceso a archivos de sistemas que manejan Windows. Sin embargo, tanto el cliente como el servidor están solo disponibles en las versiones diseñadas según Microsoft, como de uso profesional o empresarial, mientras que en las versiones antiguas (98, ME) o caseras (XP Home, Vista Basic), no es posible usar ni cliente ni servidor NFS, a menos que este provenga de un fabricante distinto, lo cual implica un costo adicional en la compra de software como NFS de Intel.

Los posibles escenarios en los que la compartición de directorios mediante NFS se podría aplicar son casi infinitos. He aquí unos ejemplos típicos:

- Tenemos una red con varias estaciones de trabajo y queremos que todas ellas posean el mismo software y la misma configuración: exportamos mediante NFS los directorios /usr (para los programas) y /etc para la configuración desde un servidor común y conseguiremos una red uniforme.
- Necesitamos tener una única copia de los archivos de un proyecto: creamos dicho proyecto en un directorio de un servidor y exportamos ese directorio a todas las estaciones de trabajo desde las que se use.
- Tenemos clientes con poco espacio en disco y necesitan correr aplicaciones grandes: se comparten los directorios de dichas aplicaciones desde un servidor mediante NFS y podremos ejecutarlas localmente en cada cliente sin problemas.
- Almacenar centralizadamente los home de n grupo de usuarios.

Puesto que tanto NFS como Samba son los estándares de hoy en día para la interoperatividad de los sistemas de archivos entre computadoras, es importante notar que los servidores UNIX/Linux pueden operar con ambos protocolos con rapidez y eficiencia, mientras que los servidores de Microsoft Windows constantemente pierden en las pruebas comparativas y no pueden proveer servicios NFS sin software y licencias adicionales.

2.3 SERVICIOS DE DIRECTORIO

2.3.1 LDAP

LDAP (*Lightweight Directory Access Protocol*), implementa un servicio de directorio jerárquico y distribuido para acceder a depósitos de información referente a usuarios, contraseñas y otras entidades en un entorno de red, ofreciendo la capacidad de filtrado sobre la información que esta siendo solicitada.

LDAP desde la trinchera de Windows representa un gran apoyo en redes muy grandes donde existen muchos usuarios, PC's, impresoras, etc. Ya que ayuda como un directorio, pero al no estar integrado en Windows, no es transparente para el usuario, ni tampoco resulta muy sencillo de usar.

Implementaciones avanzadas de LDAP como las encontradas en la arquitectura de UNIX/Linux pueden soportar decenas de directorios, mientras que en instalaciones UNIX/Linux en súper computadoras pueden encontrarse cientos de niveles de directorio LDAP.

Existe un proyecto de software libre llamado OpenLDAP⁶ y el cual fue desarrollado en la Universidad de Michigan.

2.3.2 Active Directory

Active Directory es un servicio de directorio para plataformas Windows que se encarga de almacenar y configurar datos sobre objetos de la red que se guardan en él y de administrar la actividad de los usuarios del dominio. Permite realizar tareas de administración y seguridad de una forma centralizada.

Active Directory integra el concepto de Internet conocido como espacio de nombres con los servicios de directorio del sistema operativo. Utiliza el protocolo compacto de acceso a directorios LDAP como protocolo central y puede trabajar más allá de los límites de los sistemas operativos al integrar numerosos espacios de nombres.

En una red en donde las maquinas clientes con plataforma Windows tengan la necesidad de ínter operar con otros directorios UNIX/Linux que están basados en LDAP, lo pueden hacer a través de Active Directory. Esto permite que las empresas puedan unificar y administrar los diversos espacios de nombres que ahora existen en las redes corporativas. Como consecuencia de ello se obtiene un cierto nivel de interoperabilidad necesario para administrar entornos heterogéneos.

⁶ <http://www.openldap.org>

2.4 BASES DE DATOS

2.4.1 ODBC

Uno de los principales retos a los que se enfrentan los programadores de software de aplicaciones empresariales, es decidir cual es la mejor forma de integrar las aplicaciones que crean con los distintos orígenes de datos disponibles. Además no solo suelen existir varios orígenes de información diferentes, sino que también esos orígenes van cambiando constantemente a medida que aparecen herramientas nuevas y se usan numerosas plataformas informáticas. Este entorno cambiante hace que las aplicaciones no funcionen correctamente o bien queden rápidamente obsoletas y sea necesario volver a escribirlas.

Las aplicaciones necesitan una forma flexible de tener acceso a la información disponible en distintas plataformas y en distintos orígenes de datos con la que puedan contar con el paso del tiempo.

Con el fin de resolver esa necesidad, la industria de software ha desarrollado una serie de tecnologías para el acceso a los datos y la información no vaya unido a un formato y un origen de datos determinados. Uno de los enfoques más conocidos y transparentes consiste en utilizar el ODBC (*Open Database Connectivity, Software de Conectividad Abierta de Bases de Datos*). En la actualidad, este software está disponible para los sistemas UNIX/Linux y Windows. Gracias a ODBC, un programador de aplicaciones puede crear ahora aplicaciones que tengan acceso a orígenes de datos de distintas plataformas y bases de datos, independientemente de la base de datos del entorno operativo en que se ejecute.

Esto permite al usuario escribir código de aplicaciones que puede utilizarse exactamente de la misma manera tanto en UNIX/Linux como en Windows. La aplicación puede conectarse con bases de datos que se ejecuten en cualquier plataforma, independientemente de la plataforma que en la que vaya a ejecutarse. Los usuarios tienen la posibilidad de emplear plataformas UNIX/Linux y Windows, según sean las necesidades de la organización.

La generación actual de software ODBC ha madurado y existen muchos programadores con experiencia en este software. Además, de la tecnología de software tiene un buen rendimiento en una gran variedad de bases de datos.

2.5 AMBIENTES VIRTUALES

Poder utilizar aplicaciones de otros sistemas operativos desde nuestro propio entorno cada vez es más común, ahora incluso es posible hacerlo con herramientas de virtualización.

Las computadoras modernas son suficientemente potentes y sobradas de recursos como para hacer posible y viable la virtualización; esta técnica permite la ilusión de varias máquinas virtuales más pequeñas (*vm's*) cuando solo tenemos una real. Cada uno corre una instancia de sistema operativo separado, no necesariamente el mismo del anfitrión. Sin embargo la planeación para la ejecución concurrente de sistemas operativos múltiples plantea varios desafíos.

En primer lugar, las máquinas virtuales se deben aislar mutuamente: no es aceptable que la ejecución de una afecte el correcto funcionamiento de las otras. En segundo lugar, los gastos indirectos del funcionamiento de la virtualización deben ser pequeños, o al menos menores que el usar equipo real.

VMware⁷ es una de las opciones más conocidas para la virtualización, además de ser estable y gratuito en ciertas versiones. Con una instalación simple y configuración sencilla, es posible crear una máquina virtual en nuestra computadora haciendo que sea posible ejecutar sistemas operativos iguales o distintos al principal del sistema.

La creación de máquinas virtuales toma cada vez mayor protagonismo como forma de migrar fácilmente plataformas y aplicaciones, además cómo método válido para consolidar servidores.

VMware tiene una licencia propietaria, la cual pertenece a VMware Inc. Está disponible a modo de evaluación por treinta días en ciertas versiones, predominantemente empresariales, pero es gratuito aunque sin acceso al código, en aquellas versiones básicas como VMware Server / VMware player, las cuales tienen funcionalidad completa y son actualizadas constantemente.

Las características más importantes de VMware son las siguientes:

- El sistema operativo huésped tiene acceso a todos los dispositivos de la máquina real; es decir, accede al hardware, pero a través del sistema operativo host (máquina donde corre el servidor de vmware). De esta manera el sistema operativo huésped nunca podrá corromper, por ejemplo, nuestro sistema de archivos de Linux, a pesar de que el cliente fuera un Windows infectado por virus.
- La máquina virtual puede configurarse para ser totalmente independiente en la red; es decir las demás PC's de la red la ven como si fuera una máquina real con una IP independiente, y esto permite por ejemplo, interactuar huéspedes y host para compartir archivos e impresoras entre ellos o que el sistema operativo huésped acceda a Internet usando como gateway al sistema operativo Host.
- Se pueden crear múltiples máquinas virtuales con una sola instancia de VMware y correrlas simultáneamente, limitado claro, a los recursos de hardware disponibles.

⁷ <http://www.vmware.com/>

-
- Las máquinas virtuales son simplemente un archivo en la maquina hosts y su tamaño se incrementa mientras el espacio ocupado en el disco duro virtual del sistema operativo huésped aumenta.
 - Los sistemas operativos huésped son totalmente independientes del hardware del host, por lo tanto se pueden portar fácilmente máquinas virtuales a otros hosts.

Existen además otras opciones que pueden ser muy útiles en casos donde una licencia cerrada no sea suficiente, o se tema un cambio de la estrategia de VMware donde primero el software es regalado para posteriormente forzar a algún contrato o pago.

Bochs⁸, por ejemplo es un proyecto Open Source capaz de emular una computadora completa basada en la arquitectura x86. Esto quiere decir que podemos ejecutar un sistema operativo MS-DOS o similar.

Bochs es altamente portable por estar escrito en C++. Actualmente Bochs se puede compilar para emular un 386, 486, o un Pentium. Bochs es capaz de emular la mayoría de los sistemas operativos para la arquitectura x86, entre ellos DOS, Windows 95/98, Windows NT/2000/XP. Bochs fue escrito por Kevin Lawton.

Bochs hace un trabajo muy bueno en cuanto a la emulación de una computadora x86; sin embargo en comparación con otras aplicaciones comerciales (como vmware) resulta demasiado lento y complejo tanto en la instalación y la configuración. El código fuente de Bochs es la base del proyecto plex86 que es otra alternativa libre.

Por otro lado, en el campo del software propietario, Virtual PC⁹ es una solución de virtualización que permite ejecutar sistemas operativos de PC simultáneamente en un equipo de trabajo, proporcionando una opción para mantener la compatibilidad de aplicaciones heredadas mientras se migra a un nuevo sistema operativo. Virtual PC para Mac permite ejecutar aplicaciones basadas en Windows, acceder a redes Windows, usar aplicaciones de Internet propias de Windows, y compartir archivos con otros usuarios de Windows a los usuarios de Macintosh.

Xen¹⁰ es un monitor de maquina virtual para x86 que permite la ejecución de varias instancias de un sistema operativo. Ofrece una separación total de los entornos de maquina virtual con un mínimo descenso de rendimiento. Con Xen se pueden consolidar servidores para simplificar la gestión, reducir los gastos de hardware y gestionar fácilmente cargas de trabajo variables. Para la ejecución de Xen es necesario un procesador de varios núcleos o al menos con hyperthreading. Esto no implica que sus requisitos sean muy elevados, sino que su paradigma es diferente. En lugar de proveer una capa intermedia entre un sistema operativo host y otro invitado, la cual traduce las peticiones y respuestas para que el sistema operativo invitado 'piense' que está en un equipo individual, Xen ejecuta varios sistemas operativos al mismo tiempo, sin capa de traducción, de tal forma que se aprovecha más bien las características del microprocesador para usarlo al máximo.

⁸ <http://bochs.sourceforge.net/>

⁹ <http://www.microsoft.com/spain/windows/virtualpc/default.msp>

¹⁰ <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/index.html>

Xen es una tecnología de código abierto, lanzado bajo términos de la licencia de GNU (Public Licence) Además de Linux, los miembros de la comunidad de usuario de Xen han contribuido o están trabajando en ports a otros sistemas operativos tales como NetBSD (Christian Limpach), FreeBSD (kilolibra Macy) y plan 9 (Ron Minnich). Un port de Windows XP fue desarrollado para una versión anterior de Xen, pero no esta disponible para el lanzamiento debido a las restricciones de la licencia.



CAPITULO III
SUBSISTEMAS UNIX

3.1 Tecnología Microsoft Services for Unix, SFU

SFU¹¹ (*Services for Unix*, Servicios para UNIX) cuya versión 3.5, es un subsistema basado en UNIX y que se ejecuta sobre el núcleo de Windows y permite la ejecución de programas y scripts de UNIX/Linux de forma nativa en Windows, junto con las propias aplicaciones Windows. SFU incluye más de 300 utilidades y herramientas, SFU proporciona un rango de servicios multiplataforma para interoperar Windows en entornos UNIX/Linux existentes.

SFU 3.5 ofrece una significativa mejora con respecto a la versión 3.0 en cuanto rendimiento de herramientas y servicios en entornos mixtos, en capacidades de administración de líneas de comando UNIX/Linux y la capacidad de ampliar las aplicaciones basadas en UNIX/Linux en un entorno de servicios Web Microsoft .NET. SFU dota a Windows de un completo conjunto de características adicionales que permiten una mayor interoperabilidad con los servidores UNIX / Linux existentes en la infraestructura instalada.

Dichas características se encuadran en tres categorías:

- Servicios de archivos: Compatibilidad con el cliente de Sistema de archivos de red (NFS) y con el servidor NFS, que permite usar los recursos del sistema de archivos de UNIX/Linux y que los clientes UNIX/Linux puedan utilizar los recursos del sistema de archivos de Windows.
- Servicios de conectividad: servidor y cliente telnet mejorado basado en texto, además de un demonio de sincronización de contraseñas que proporciona un único punto de administración de contraseñas para sistemas basados en Windows y en UNIX/Linux.
- Servicios de facilidad de uso: utilidades para UNIX/Linux y un shell Korn para proporcionar al usuario y al administrador de UNIX/Linux un subconjunto conocido de las utilidades y las características de shell que la mayoría de usuarios de UNIX/Linux conocen y utilizan.

El sistema de archivos en Red NFS (*Network File System*) ofrece una significativa mejora en el rendimiento con respecto a sus versiones anteriores de SFU, permitiendo a los usuarios acceder a archivos en forma sencilla en máquinas que funcionan tanto bajo Windows como UNIX.

A través de la sincronización de contraseñas de doble vía mejorada, correspondencia de nombre de usuario y el servidor NIS (*Network Information Service*), se pueden integrar o centralizar los servicios de directorio a través de las plataformas UNIX/Linux y Windows con el servicio Active Directory.

SFU 3.5 incluye herramientas para administrar los entornos mixtos de manera simple e incluye una serie de herramientas basadas en UNIX/Linux para administrar la plataforma Windows, lo que permite a las compañías aprovechar la experiencia adquirida en UNIX/Linux por sus administradores y aplicarla a la plataforma Windows. Además de las capacidades de registro dinámico, que permitirán a los administradores de redes hacer cambios como ajustes de la red, sin que el sistema tenga lapsos de caída, que antes en Windows solo se resolvían reiniciando la máquina.

¹¹ <http://www.microsoft.com/windows/sfu/>

Con el uso de SFU se aprovechan las inversiones en sistemas UNIX actuales con el kit de desarrollo de software y subsistema Interix, el cual es un completo sistema de ejecución de aplicaciones, que permite a los clientes recopilar y poner en funcionamiento de forma nativa programas y scripts UNIX/Linux sobre sistemas operativos Windows. Utilizando Interix se reutilizan las aplicaciones UNIX/Linux integrándolas con la plataforma Windows. Las capacidades de Interix incluyen soporte para aplicaciones en entornos mixtos, lo que permite poner en funcionamiento sobre plataforma Windows un mayor número de aplicaciones que hayan sido desarrolladas para UNIX/Linux.

El subsistema SFU aporta herramientas de interoperabilidad entre Windows y UNIX/Linux. SFU busca reducir costos y mejorar la eficacia de los recursos tecnológicos, proporcionando una amplia gama de servicios en entornos mixtos que interoperan Windows y UNIX.

SFU 3.5 proporciona un solo paquete para satisfacer los requisitos de interoperabilidad con las características siguientes:

- **Componentes del Sistema de Archivos NFS (Network File System):**
 - Soporta la interoperatividad de archivos y directorios entre sistemas UNIX/Linux y Win.
 - **Cliente para NFS:** permite a las computadoras de Windows acceder a archivos en servidores NFS basados en UNIX/Linux.
 - **Servidor para NFS:** Exporta directorios de Windows como sistemas de archivos NFS, permitiendo que los clientes UNIX/Linux basados en NFS tengan acceso. Cabe mencionar que el servidor NFS soporta los exports NFS desde sistemas de archivo CDFS y NFTS únicamente. No están soportados los sistemas FAT y FAT32.
 - **Gateway para NFS:** Comparte los directorios exportados por los servidores UNIX/Linux NFS como directorios compartidos de Windows, permitiendo que los usuarios de Windows tengan acceso sin la instalación del software del cliente NFS.

- **Herramientas de autenticación para NFS:** permite que los componentes del NFS autenticuen usuarios.
 - **Mapeo de nombres de usuario:** Asocia nombres de usuario de Windows y sistemas UNIX/Linux para permitir que los usuarios tengan acceso a archivos con Interix o para conectar con los recursos del NFS sin tener que entrar a los sistemas de UNIX/Linux y Windows por separado.
 - **Servidor para la autenticación del NFS:** Trabaja con el mapeo de nombres de usuario y servidor para NFS para permitir a los usuarios UNIX/Linux el acceso transparente a archivos compartidos en el servidor para NFS.

- **Acceso remoto a línea de comando entre UNIX/Linux y Windows.**
 - Cliente Telnet
 - Servidor Telnet

- **Capacidades de script de plataforma cruzada.**

-
- Korn Shell
 - C shell
 - Más de 350 comandos UNIX y utilidades.
 - Vínculos simbólicos y duros en sistemas de archivos NTFS y NFS.
- **Administración común de la red proporcionando funcionalidad de un servidor NIS Usando el servicio de Active Directory**
 - **Sincronización de contraseñas entre Windows y UNIX.**
 - **Instalación usando Microsoft Windows Installer**
 - **Administración simple de los componentes y servicios de SFU** Usando Microsoft Management Console (MMC) o una línea de comandos.
 - **Instalable en diferentes versiones de Windows.** Instalación en computadoras corriendo Windows 2000, Windows XP Professional y Windows Server 2003¹².
 - **Compatibilidad con varias versiones y distribuciones de UNIX** – probados específicamente en Solaris 7 y 8, HP-UX 11i, RedHat Linux 8.0; y IBM AIX 5L .2.

Windows Services for Unix incluye los siguientes componentes:

- **Interix SDK:** Un kit de desarrollo de software (SDK), que permite crear aplicaciones mas fácilmente para correr en un subsistema Interix. Instalando este componente también se instalan las utilidades básicas.
- **Componentes de Conectividad Remota:** Permite a los usuarios acceder a una computadora de forma remota.
 - **Servidor Telnet:** proporciona a los usuarios remotos la capacidad de conectar con el servidor con cualquier software de cliente telnet.
 - **Servicio Remoto de escritorios Windows:** permite que los comandos de las computadoras remotas funcionen en el servidor.
- **Utilerías:** Colección de utilerías comunes en estaciones UNIX/Linux.
 - **Utilerías Base:** provee el subsistema Interix, el cual es un ambiente UNIX/Linux completo en características, que se ejecuta por separado como un subsistema de Windows. Incluye el compilador C y terminales Korn, además de 350 utilerías que se ejecutan en interix. Además este componente provee utilerías de administración basadas en Windows y comandos al estilo UNIX/Linux, además de un Cron basado en Windows que calendariza tareas, además un cliente Telnet.
 - **Perl de UNIX:** permite que los scripts de perl funcionen en el subsistema Interix. La instalación de Perl de UNIX/Linux también instala las utilidades básicas.
- **Componentes de Interix GNU:** Herramientas de desarrollo, utilidades y software de Interix que son distribuidas bajo términos GNU.

¹² Service Pack 4 es requerido en Windows 2000 y service pack 1 es requerido en Windows XP.

-
- **Utilidades de Interix, GNU:** Suple las utilerías básicas. La instalación de las utilerías del GNU también instala las utilidades básicas.
 - **Interix GNU SDK:** Herramientas del kit de desarrollo del software (SDK). La instalación de las herramientas del GNU SDK
 - **Sincronización de contraseña:** Simplifica a los usuarios mantener una contraseña para los dominios de Windows y sistemas UNIX/Linux, sincronizándolos cuando uno de ellos cambie.

SFU proporciona un conjunto de herramientas para tender un puente ente sistemas UNIX/Linux y Windows, tanto en usuarios como en administradores. Con SFU, es posible crear una red empresarial, lógica y consistente, donde los recursos se comparten sin cambios y donde el control de acceso es determinado por las políticas de la empresa en lugar de la plataforma.

SFU 3.5 aporta ventajas como las siguientes:

- Compartir de la misma manera datos entre servidores y clientes corriendo cualquier sistema operativo ya sea UNIX/Linux o Windows.
- Acceder remotamente a la línea de comando a las computadoras de UNIX/Linux o de Windows desde cualquier computadora con sistema operativo Windows o UNIX.
- Hacer scripting y desarrollo de aplicaciones. La tecnología del subsistema Interix proporciona unos escritorios y utilidades familiares UNIX/Linux junto con todo el sistema completo de las APIs de UNIX/Linux.
- Administración de red heterogénea, incluyendo la sincronización de contraseñas.
- Simple instalación integrada.
- Fácil de utilizar.

El 14 de enero del 2004 en Framingham, Mass., se llevó a cabo la exposición para los premios *Linux World Product Excellence*¹³. En esta, la plataforma Services for Unix 3.5 fue nominada como finalista dentro de la categoría de Mejor Solución de Integración.

Uní forum Association, en conjunto con los premios de *Linux World Product Excellence Awards* son eventos de premiación a productos y servicios de software reconocidos por los participantes de la convención Linux World. Dichos premios están divididos en 10 categorías que representan las áreas más representativas en innovación Linux y la comunidad Open Source. Los jueces seleccionaron en este caso, un producto o servicio que represente el avance más significativo para Linux en el mercado

SFU puede ser descargada en la siguiente dirección:

<http://www.microsoft.com/windows/sfu/downloads/default.asp>.

¹³ <http://www.linuxworldexpo.com/live/12/media/news/CC713169>

3.2 Tecnología CygWin

Cygwin¹⁴ es un entorno UNIX/Linux completo utilizable en plataformas Windows. Se trata de una biblioteca de enlace dinámico (DLL, por sus siglas en inglés) que emula este entorno de trabajo y, que va acompañado de multitud de paquetes UNIX/Linux portados para ser utilizados.

Esta herramienta fue desarrollada por Cygnus Solutions para proporcionar un comportamiento similar a los sistemas UNIX/Linux en Windows. Su objetivo es portar software UNIX/Linux a Windows que cumpla con el estándar POSIX¹⁵. Aunque los programas portados funcionan en todas las versiones de Windows, su funcionamiento es obviamente mejor en Windows NT, Windows XP y Windows 2003 Server, ya que su arquitectura fue inicialmente orientada hacia la red, a la multitarea, a la existencia de procesos, multihilos, etc.; características no disponibles en las versiones rudimentarias como Windows 95/98/ME.

Cygwin es una capa intermedia que convierte las llamadas POSIX a las llamadas de la API nativa de Windows. Además permite que las aplicaciones POSIX “vean” un sistema de archivos igual al de los sistemas UNIX/Linux.

El sistema Cygwin tiene varias partes diferenciadas:

- Una biblioteca DLL (cygwin.dll) que implementa la API POSIX usando para ello llamadas a la API nativa de Windows Win32.
- Una cadena de desarrollo GNU (que incluye entre otras utilerías GCC y GDB) para facilitar las tareas básicas de desarrollo.
- Aplicaciones equivalentes a los programas más comunes de los sistemas UNIX/Linux. Incluso incluye un sistema X (cygwin/X).

En la siguiente figura se han esquematizado las partes de Cygwin.

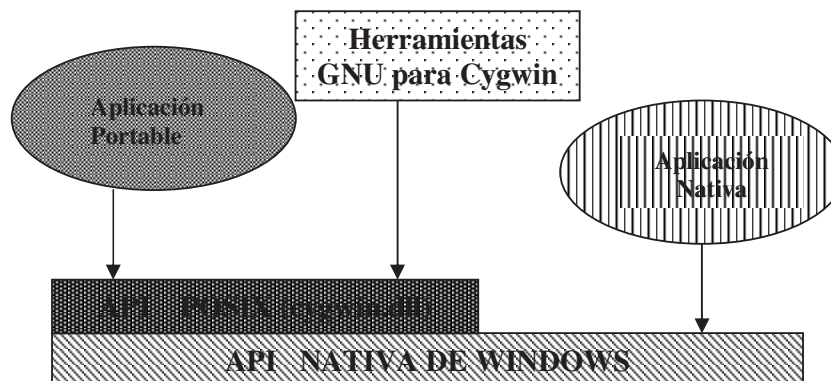


Fig. 3 Esquema CygWin

La parte cygwin.dll es la que ofrece la capa POSIX: cualquier aplicación que utilice exclusivamente esta API, será una aplicación portable. Cualquier aplicación que use la aplicación API nativa de Windows, no lo será. Para compilar las aplicaciones portables desde una maquina Windows, se necesitan las herramientas GNU para cygwin.

¹⁴ <http://www.cygwin.com/>

¹⁵ Portable Operating System Interface, proviene la X de UNIX.

Cygwin viene con un instalador mínimo que nos permite personalizar los paquetes a bajar, Cygwin se encarga del resto. Recién instalado, ya es posible hacer uso del bash (línea de comandos de UNIX/Linux), correr scripts de perl que solo corren bajo UNIX/Linux, servidores que solo se consiguen en forma de código fuente, usar Cron para tareas recurrentes y ncftp y wget para bajar archivos. En fin, existen muchos programas para Linux que pueden ser ejecutados sobre Cygwin. Todo esto desde el mismo Windows.

El shell (bash de Linux) permite trabajar en maquinas Windows como si de sistemas UNIX/Linux se tratase.

Además, existe una larga lista de aplicaciones¹⁶ que corren en Cygwin. En la actualidad, el software esta mantenido principalmente por RedHat. Se distribuye bajo los términos de la GPL¹⁷ por lo que permite ser enlazada con cualquier tipo de software cuya licencia este de acuerdo con la definición de software libre.

SFU (*Services for Unix*) es un paquete que incluye todo lo disponible -esto significa también que no hay disponible nada adicional-, convirtiéndolo en un instalador bastante grande, mientras que Cygwin es un pequeño instalador con lo básico, el cual es mantenible y extensible a través de Internet, y cuya instalación completa pueden hacerlo llegar a mas de un gigabyte en disco incluyendo finalmente, muchas mas utilerías y software que un paquete completo de SFU, además de que dicha colección va creciendo y mejorando.

3.3 UWIN una alternativa a CygWin

El subsistema Uwin¹⁸ es otra alternativa de software abierto de los laboratorios AT&T que permite construir y ejecutar, con pocas o ninguna modificación, aplicaciones UNIX/Linux en Windows desde Windows 95 hasta XP.

En Uwin la API de UNIX/Linux esta implementada con la librería dinámica POSIX.DLL, y además provee las herramientas necesarias para compilar en Windows las aplicaciones UNIX usando los compiladores Visual C/C++5.x,6.x y 7.x, BorlandC/C++, Mingw y Digital Mars C/C++, además por supuesto el compilador GNU.

Uwin contiene:

- Bibliotecas que emulan el entorno de UNIX implementando la API¹⁹.
- Incluye los archivos y las herramientas de desarrollo como: cc, yacc, lex, y make.
- Ksh (escritorio Korn) y mas de 250 utilidades como: ls, sed, cp, stty, etc.

La mayoría de la API de UNIX esta implementado por la librería POSIX.DLL, cargadas dinámicamente. Los programas se ligaron a POSIX.DLL corren bajo el subsistema Win32 en vez del subsistema POSIX, así que los programas pueden entremezclarse libremente UNIX y llamadas a la librerías Win32.

¹⁶ http://www.hirmke.de/software/develop/gnuwin32/cygwin/porters/Hirmke_Michael/GNUWin32-links.html

¹⁷ <http://www.gnu.org/licenses/licenses.es.html>

¹⁸ <http://www.research.att.com/sw/tools/uwin/>

¹⁹ Application Programming Interface

Uwin funciona mejor en Windows XP/NT/2000 con NTFS, pero también trabajara en un modo degradado en un sistema de archivos FAT y con aun más de degradación bajo Windows ME/95/98.

Uwin proporciona la funcionalidad siguiente para Win32:

- Control y gestión de procesos:

Mientras que los procesos pueden ser creados usando la función *fork* (2), Uwin incluye una familia de funciones *spawn* que combinan funcionalidad de *fork/exec* para la eficiencia. Cada proceso tiene un identificador de proceso único y cada uno pertenece a un grupo de procesos. La función *vfork* es también una manera eficiente de crear procesos.

- Descriptores de Archivo

Uwin Abre archivos, pipes, sockets, fifos y caracteres y bloques especiales; los archivos de dispositivos tienen descriptores de archivo asociados a ellos, y se heredan con la semántica de UNIX.

- Señales de UNIX/Linux:

Se proporcionan casi todas las señales de UNIX/Linux, incluyendo control de señales, de modo que *ksh* puede parar y recomenzar trabajos. Un proceso puede recibir, bloquear, o no hacer caso de señales. Las señales pueden enviar a procesos o a grupos de procesos.

- Soporte de dispositivos:

Uwin provee dispositivos como los encontrados en los sistemas UNIX/Linux. Provee Soporte para acceso directo a los dispositivos del floppy (*/dev/fd0*), y dispositivos SCSI (*/dev/mt0*) así como *ptys* y *ttys*.

- Terminal para consolas, Sockets, y líneas seriales:

Se soportan consolas con la emulación de VT100. Líneas seriales, y sockets están diseñadas como terminales virtuales.

- Uso del ratón con las ventanas de la consola:

El botón izquierdo del ratón en cada ventana de la consola se puede utilizar para seleccionar el texto y copiar al portapapeles. El botón derecho (o botón medio en ratones de tres botones) puede ser utilizado para pegar el texto del portapapeles.

- Sockets UCB (*University of California at Berkeley*) basados en WinSock

Los sockets utilizan los archivos de encabezado y la convención de nombres de UCB, pero son implementados como llamadas a WinSock. Los sockets son descriptores de archivo y obedecen la semántica de descriptores de archivos.

- Rutas de directorio de UNIX/Linux a Windows

Uwin provee el estilo de UNIX/Linux para nombrar los archivos y usa “/” como delimitador y en la creación de montajes. La localización del directorio raíz / se puede seleccionar durante la instalación. Uwin también soporta nombres UNC (*Universal Naming Convention*) para los archivos que comienzan con “//”.

- Convenciones de Nombres de UNIX/Linux.

Por defecto los directorios son montados sin distinguir mayúsculas de minúsculas. Sin embargo el comando *mount* permite a los directorios ser montados como sensible a minúsculas de modo que los archivos *makefile* y *Makefile* sean distintos.

- Mapeo de permisos UNIX/Linux a permisos de NT/2000/XP

Los identificadores de Windows NT/2000/XP son mapeados a los identificadores de usuario y de grupo de UNIX/Linux. Los permisos de UNIX/Linux son mapeados en Windows NT/2000/XP mediante los archivos ACLs. Se puede utilizar *chown* () para cambiar al dueño y/o grupo de un archivo.

- Mapeo de la memoria y memoria compartida

Se proporcionan *mmap* () y las capacidades de compartición de memoria.

- Vínculos duros

Los vínculos duros se soportan en los sistemas de archivos de NTFS y FAT.

- Vínculos Simbólicos

Los vínculos simbólicos a los archivos y a los directorios se pueden crear dentro de Uwin y son como atajos de Windows. El sufijo *.ink* es añadido al nombre de archivo pero no aparece al leer directorios con Uwin. Los atajos creados por Windows aparecen como los vínculos simbólicos en Uwin pero no tienen el sufijo *.ink*

- Acceso al registro de Windows a través del sistema de archivos

Uwin trata al registro de Windows como un sistema de archivos con las llaves finales tratadas como archivos, y llaves que contienen otras llaves como directorios. El registro se monta automáticamente debajo de */reg* durante el arranque.

- Utilidades AT&T de fuente abierta

La mayoría de las utilidades de Uwin se basan en utilidades de fuente abierta de acuerdo con los estándares POSIX y X/OPEN. Las utilidades generan sus propias páginas de ayuda.

- Herramientas de desarrollo

Uwin incluye wrappers para los compiladores cc que utilizan Visual C/C++ para construir aplicaciones Uwin. Las Herramientas de desarrollo como: yacc, lex, ar, nm, y rcs también se proporcionan. Uwin proporciona ambos el tradicional *make* y *AT&T nmake*.

- KornShell

El *KornShell* (ksh) puede iniciarse con un doble clic en el icono etiquetado como “ksh para Windows NT” el *kornshell* funciona en una ventana de la consola justo como el comando de MS-DOS. Una vez que *ksh* esta funcionando, todas las utilidades de UNIX/Linux pueden ser ejecutadas. *Ksh* puede ejecutar aplicaciones nativas de Windows. Algunas variables como *PATH* que son interpretadas de forma distinta por ambos Windows y UNIX/Linux ya que utilizan distintos formatos convertidos a formatos UNIX/Linux cuando se ejecutan las utilidades de UNIX/Linux, y convertidas de regreso cuando ejecutan utilidades Windows.

La documentación en línea es obtenida haciendo clic en el icono de referencia rápida de Uwin. La mayoría de los comandos generan su propia documentación cuando se invoca el comando como *command --man*. Además *man -b command* mostrara la documentación del comando en la ventana del explorador de Internet.

3.4 Tabla Comparativa entre herramientas

	Comandos y utilerías disponibles	Compatibilidad con Windows	Integración con Windows
SFU	Mas de 350	A partir de NT	Simple
Cygwin	n/d (numero muy grande, superior sin duda a cualquiera de las alternativas analizadas)	Todas pero algunas versiones con degradación	Manual
Uwin	Mas de 250	Todas pero algunas versiones con degradación	Manual

	Tamaño (Instalación completa)	Aplicaciones	Rendimiento
SFU	500Mb	Media	Medio
Cygwin	1.5Gb +	Amplias	Alto
Uwin	35Mb	Escasas	n/d

Tabla 2. Tablas comparativas entre herramientas

CAPITULO IV
IMPLANTACION, PRUEBAS Y COMPARATIVAS

4.1 IMPLANTACION

4.1.1 Services for Unix

Requerimientos del Sistema.

Los requisitos mínimos de espacio en disco para instalar Microsoft Services for Unix dependen de los componentes que se instalen y el sistema de archivos del disco donde los componentes serán instalados.

El espacio de disco máximo requerido para instalar todos los componentes de Services for Unix es aproximadamente de 360MB. Pero es recomendable reservar al menos 500MB para una instalación completa, y el mínimo espacio en disco requerido es de 21MB si solamente se instala el servidor NFS.

En la tabla siguiente se muestran los requerimientos de sistema operativo en cada componente; en la primera columna aparece el nombre del componente y entre paréntesis el espacio en disco que se requiere tener libre para instalar dicho componente, y en las siguientes columnas aparecen los diferentes sistemas operativos, si cuenta con una '✓' indica que el componente viene ya instalado o bien se puede agregar con una instalación modificada, por lo contrario si aparece la designación "N/D" indica que dicho componente no esta disponible en el sistema operativo especificado.

Componente	SISTEMA OPERATIVO			
	Windows 2000 Professional	Windows XP Professional, Windows XP Edicion Tablet PC	Servidor Windows 2000	Servidor Windows 2003
Utilerías (122 MB)	✓ ESTANDAR	✓ ESTANDAR	✓ ESTANDAR	✓ ESTANDAR
UNIX Perl (38 MB, más requisitos para las Utilerías)	✓ ESTANDAR	✓ ESTANDAR	✓ ESTANDAR	✓ ESTANDAR
Utilidades de Interix GNU (8 MB mas requisitos par alas utilerías)	✓ ESTANDAR	✓ ESTANDAR	✓ ESTANDAR	✓ ESTANDAR
Kit del desarrollo del software de Interix GNU (SDK) (56 MB, mas requisitos para utilerías GNU y SDK Interix)	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA
Servidor NFS (2 MB)	✓	✓	✓	✓

	MODIFICADA	MODIFICADA	ESTANDAR	ESTANDAR
Cliente para NFS (1 MB)	✓ ESTANDAR	✓ ESTANDAR	✓ ESTANDAR	Xì ✓ ESTANDAR
Gateway para NFS (2 MB)	N/D	N/D	✓ MODIFICADA	✓ MODIFICADA
Servidor para NIS (5 MB)	N/D	N/D	(Solo Controlador de dominio)	(Solo Controlador de dominio)
Sincronización de contraseñas (1 MB)	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA	✓ ESTANDAR
Servidor Telnet (1 MB)	✓ ESTANDAR	N/D	✓ ESTANDAR	N/D
Servicio Remoto de escritorio basado en Windows (1 MB)	✓ MODIFICADA	✓ MODIFICADA	✓ ESTANDAR	✓ ESTANDAR
Mapeo de Nombres de Usuario (1 MB)	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA	✓ ESTANDAR
Servidor para la autenticación del NFS (1 MB)	✓ MODIFICADA	✓ MODIFICADA	✓ ESTANDAR	✓ ESTANDAR
Servidor para PCNFS (1 MB)	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA
Interix SDK (41 MB, Mas requerimientos Para Utilerías)	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA
ActiveState Perl (59 MB)	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA	✓ MODIFICADA

Tabla 3. Requerimientos de Sistema Operativo

La instalación de prueba ha sido implementada sobre una versión en español de Microsoft Windows XP profesional, con un procesador Pentium IV a 1.60GHz con 256MB en RAM.

SFU viene con un instalador estándar y es recomendable reservar por lo menos 500MB para la instalación completa, de lo contrario si no se cuenta con el espacio suficiente, bastará con reservar 21MB que cuenta con la instalación básica estándar y además el servidor NFS. La instalación no tiene en realidad complejidad ya que basta con seguir el asistente con los valores que aparecen por defecto para poder ejecutar la mayoría de las aplicaciones, se muestra a continuación pasa a paso todas las ventanas de dicha instalación.



Fig.4 Ventana Inicial del programa Sfu Setup

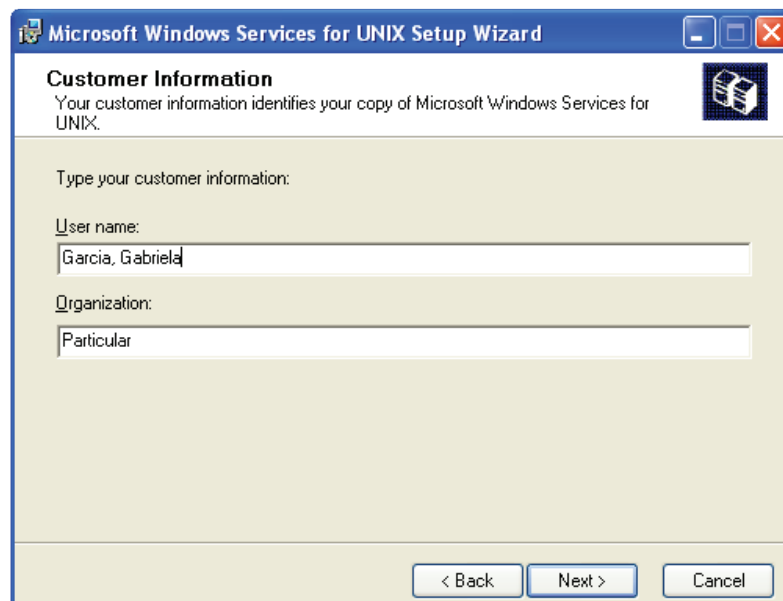


Fig.5 Ventana de Información del cliente



Fig.6 Ventana de acuerdo de Licencia

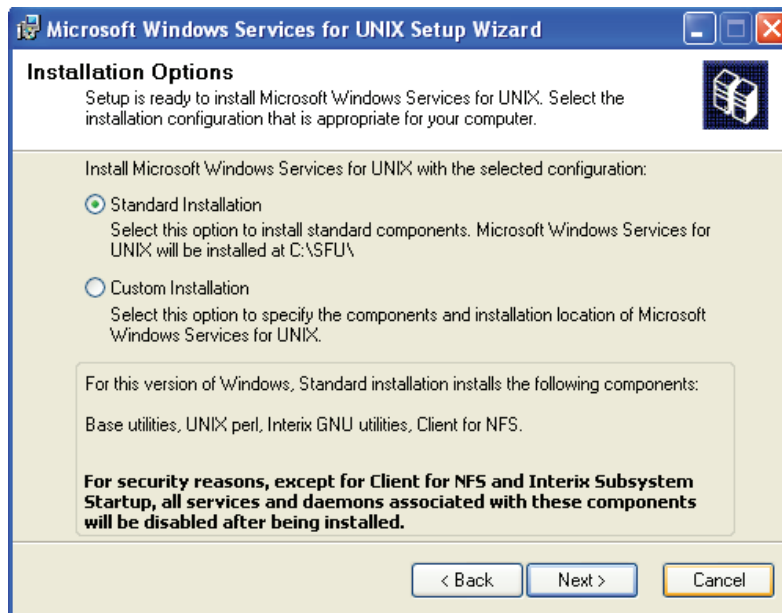


Fig.7 Opciones de Instalación.

La configuración de instalación de SFU tiene dos opciones: Una instalación estándar y una modificada, en este caso se eligió una instalación estándar que cuenta con ciertos componentes, pero en dado caso de que se requiera agregarle o quitarle componentes a la instalación se tendría que elegir una instalación modificada y de esta manera se seleccionan los componentes que se requieran.

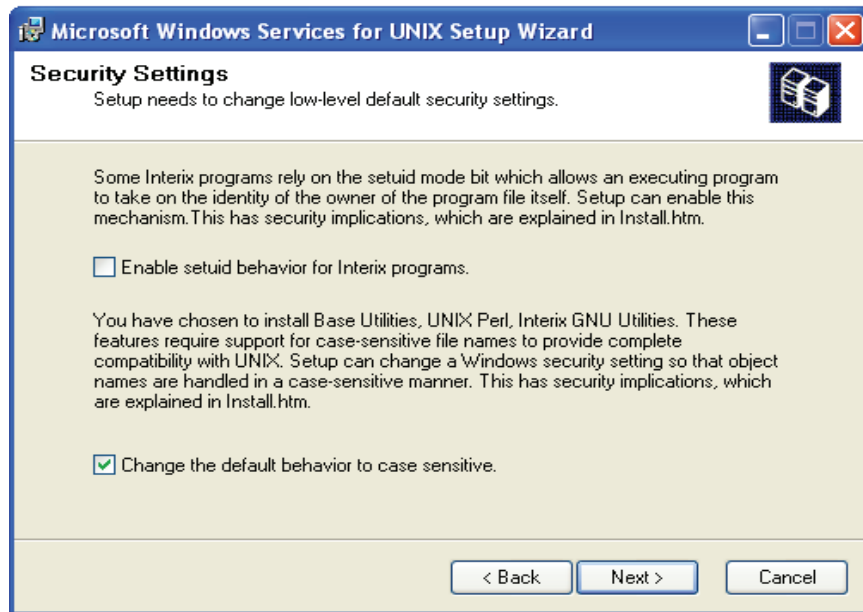


Fig.8 Ajustes de Seguridad y de sensibilidad a mayúsculas y minúsculas

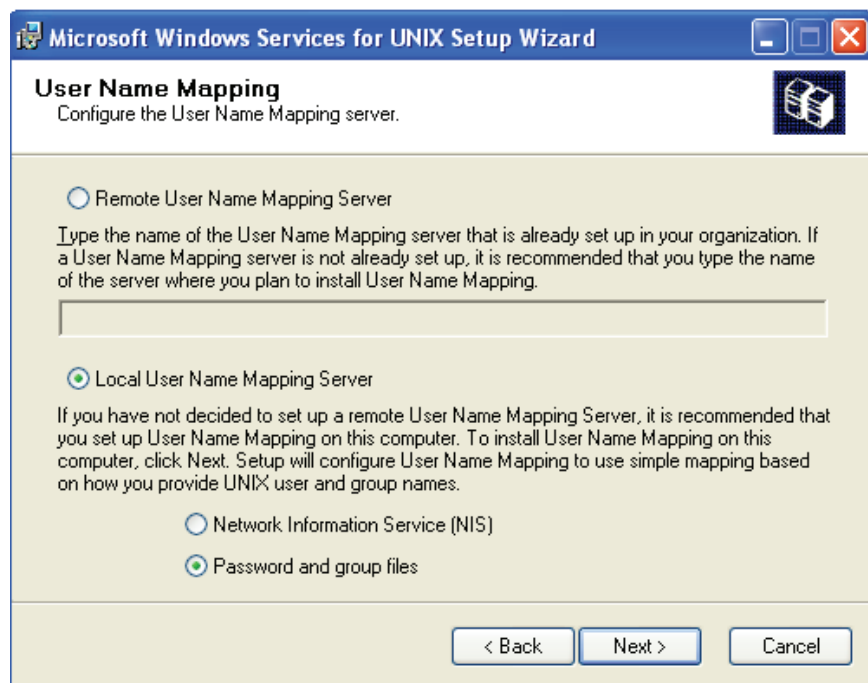


Fig.9 Configuración del Mapeo de nombres de Usuario

En este diálogo de mapeo de nombres de usuario se eligió la opción de usuario local donde se toma un archivo *passwd* de una máquina UNIX existente que además será la que usará los servicios de SFU. Con este archivo se hace la relación uno a uno con los usuarios de Windows del servidor, es decir si en la maquina UNIX el nombre de usuario es *gabriela* y en Windows este mismo usuario tiene el nombre de *gaby*, con esta relación uno a uno entre usuarios le indicaremos a nuestro servidor Windows que usuario Windows corresponde a cada usuario UNIX, por ejemplo *gabriela* = *gaby*; esta opción es la más simple para una implantación de pruebas, pero no es mantenible para un sistema en producción y por lo tanto no es la más

recomendable. Para nuestros propósitos de pruebas se eligió la opción de usuario local ya que simplifica la implantación.

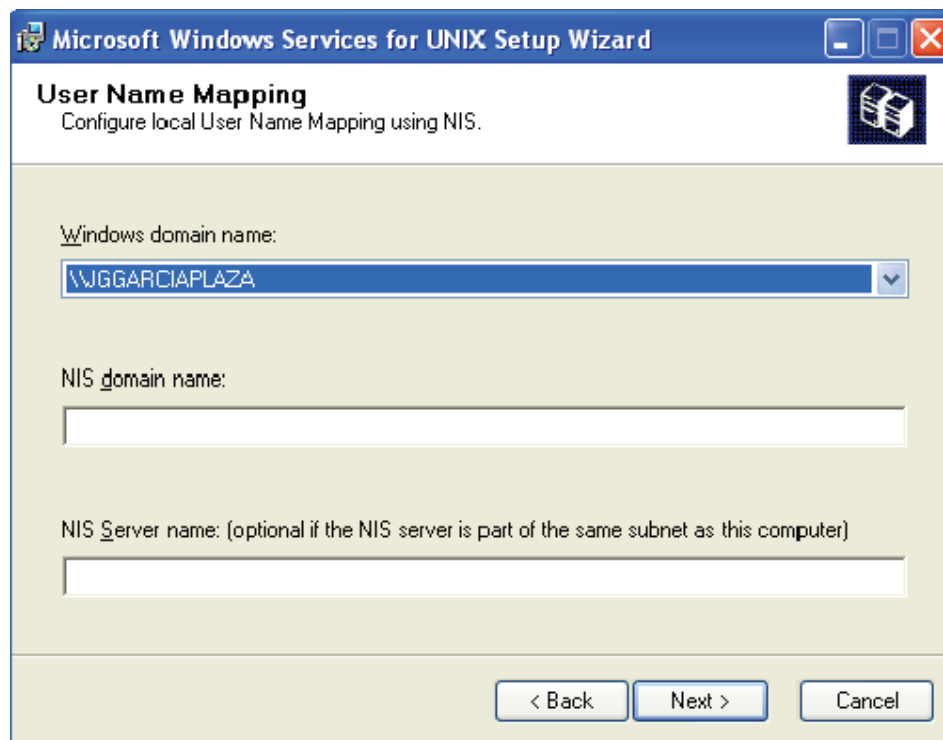


Fig.10 Configuración local del Mapeo de Usuarios con NIS

Configuración de usuario local de mapeo de nombres usando NIS (*Network Information Server*). Este dialogo aparece cuando no se usa el mapeo basado en un archivo passwd. Ya que no fue la opción seleccionada en la instalación de pruebas, se presenta solo con fines ilustrativos.

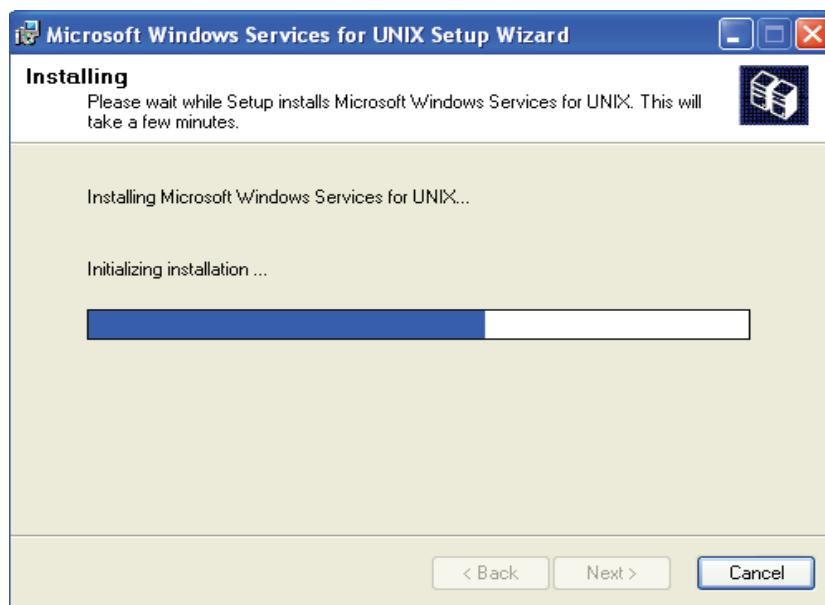


Fig.11 Dialogo de Inicio de instalación

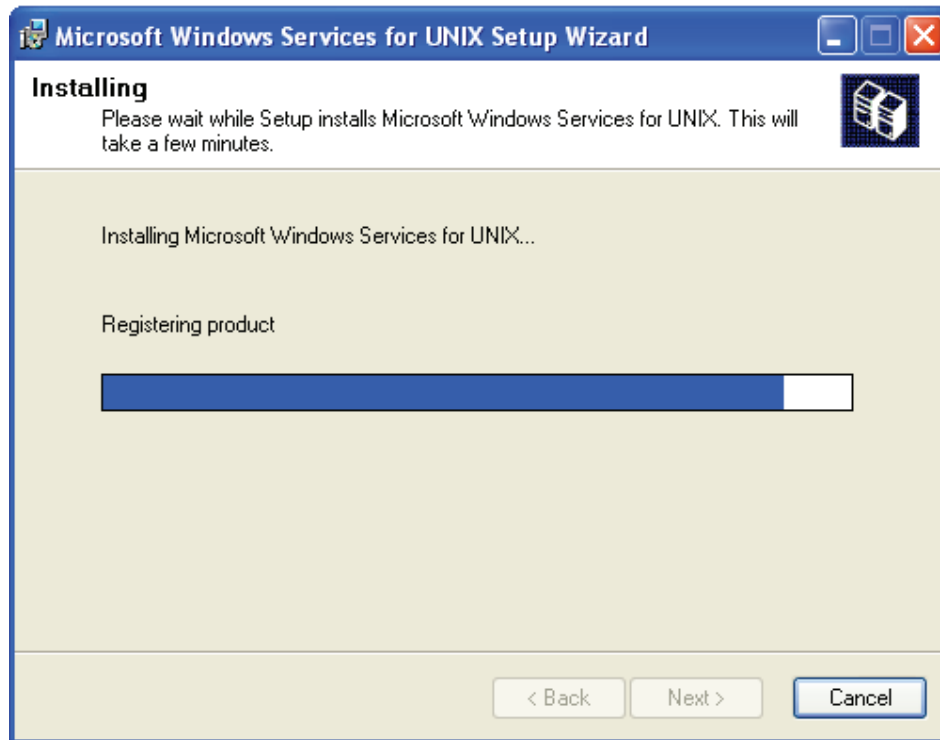


Fig.12 Barra de avance durante la instalación.

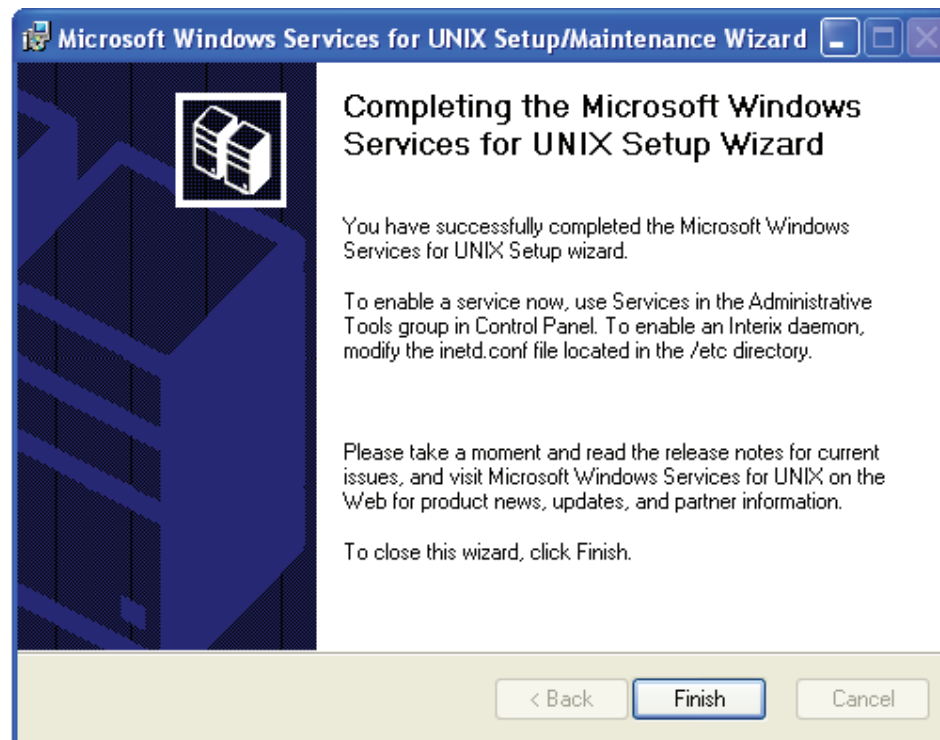


Fig.13 Instalación completa.

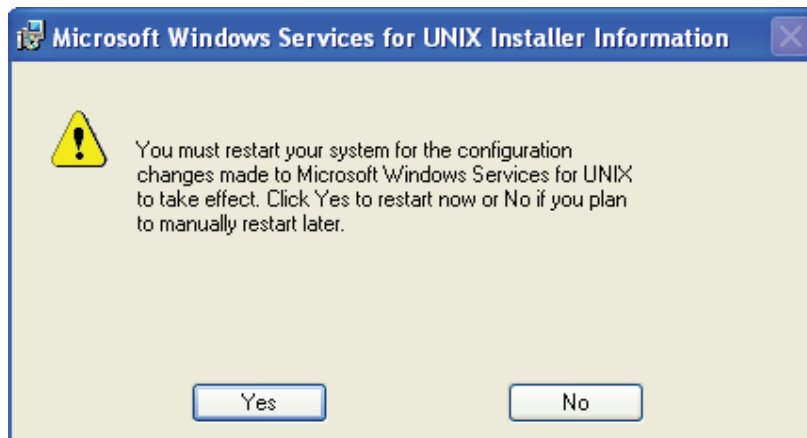


Fig.14 Reinicio del sistema.

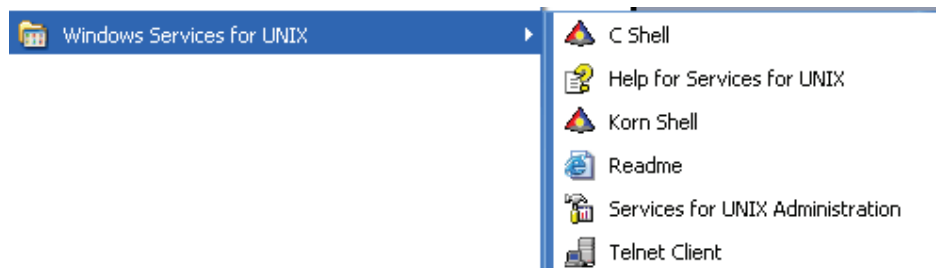


Fig.15 Aquí se muestra SFU instalado, y cuenta con diferentes opciones a elegir.

4.1.2 Cygwin

Cygwin es un subsistema completo para aplicaciones UNIX utilizable en plataformas Windows. Antes de comenzar con la instalación Cygwin, se deben tomar en cuenta los siguientes aspectos:

- Verificar que la versión de Cygwin sea reciente (por lo menos febrero 2004), ya que el software Cygwin ha evolucionado mucho y las diferencias de funcionalidades entre versiones son muy marcadas.
- El espacio en disco depende de las utilerías y paquetes seleccionados, y puede llegar hasta 1.5Gb en una instalación completa.
- Esta versión particular de instalación ha sido implementada sobre una versión en español de Windows XP home, con la versión de Cygwin 2.510.2.2 se ha probado en un sistema AMD Athlon 1.05GHz con 192MB en Ram.

El archivo de instalación de Cygwin es un archivo pequeño que realizara una descarga de Internet. Dicha descarga depende del número de los paquetes a instalar; dichos paquetes seleccionados serán descargados y copiados al disco duro. A continuación se muestran la serie de ventanas que se muestran durante su instalación.



Fig.16 Icono Inicial del programa *setup.exe*

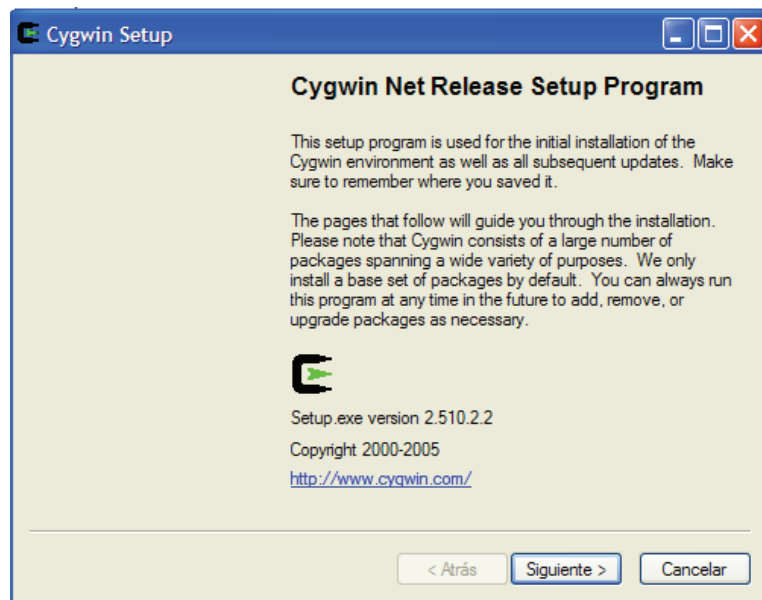


Fig.17 Selección de la operación de instalación

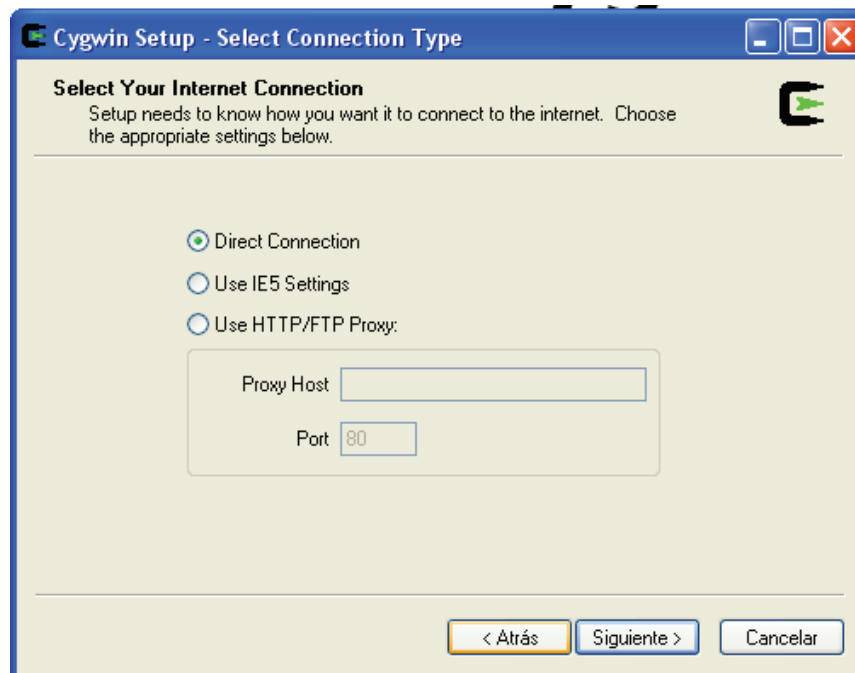


Fig.18 Tipo de conexión.

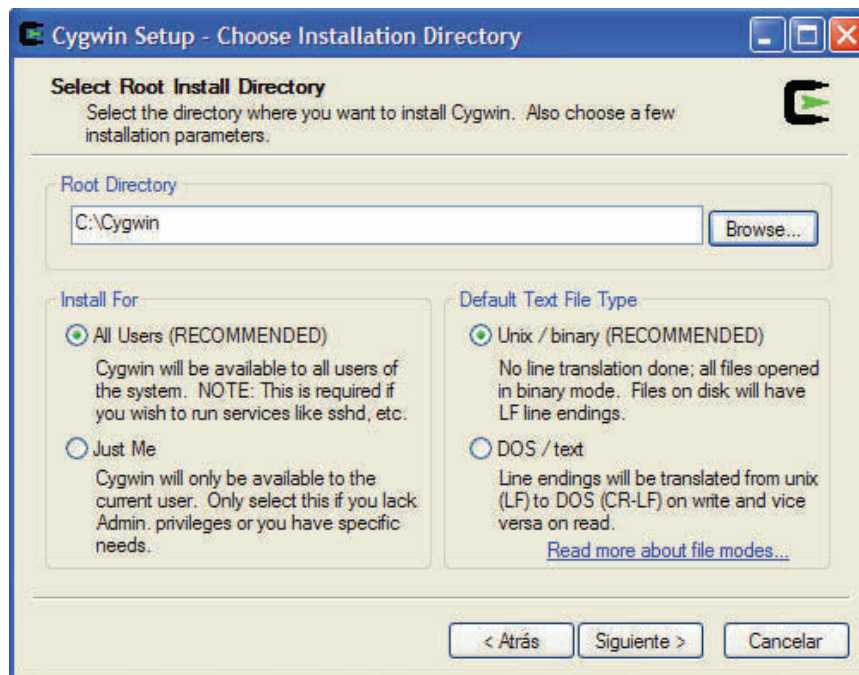


Fig. 19 Selección del directorio donde se instalara Cygwin

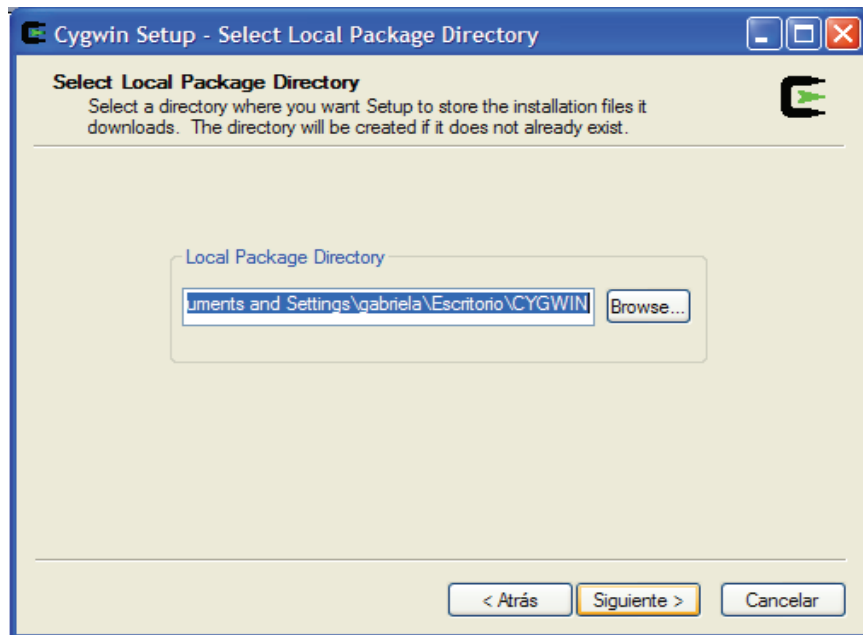


Fig. 20 Selección del directorio donde se guardarán los archivos de instalación

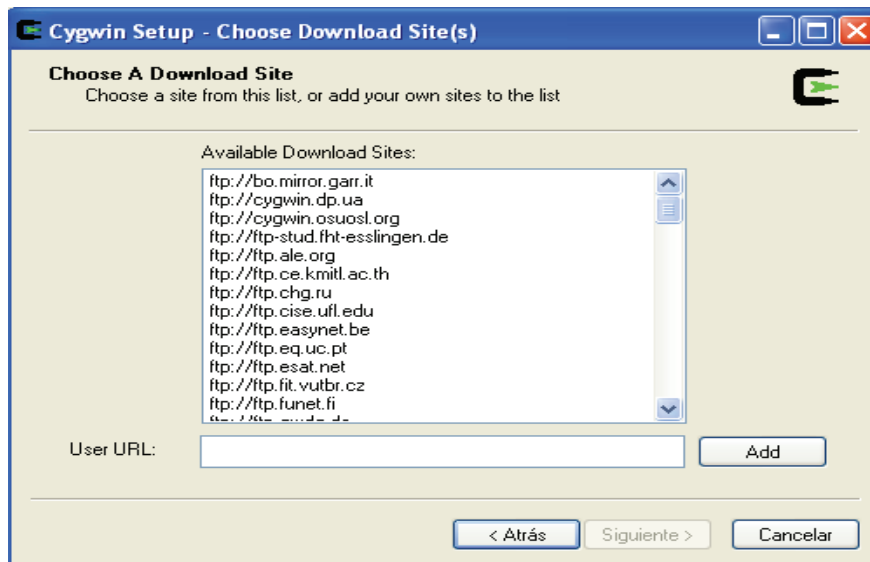


Fig.21 Selección de sitio de descarga

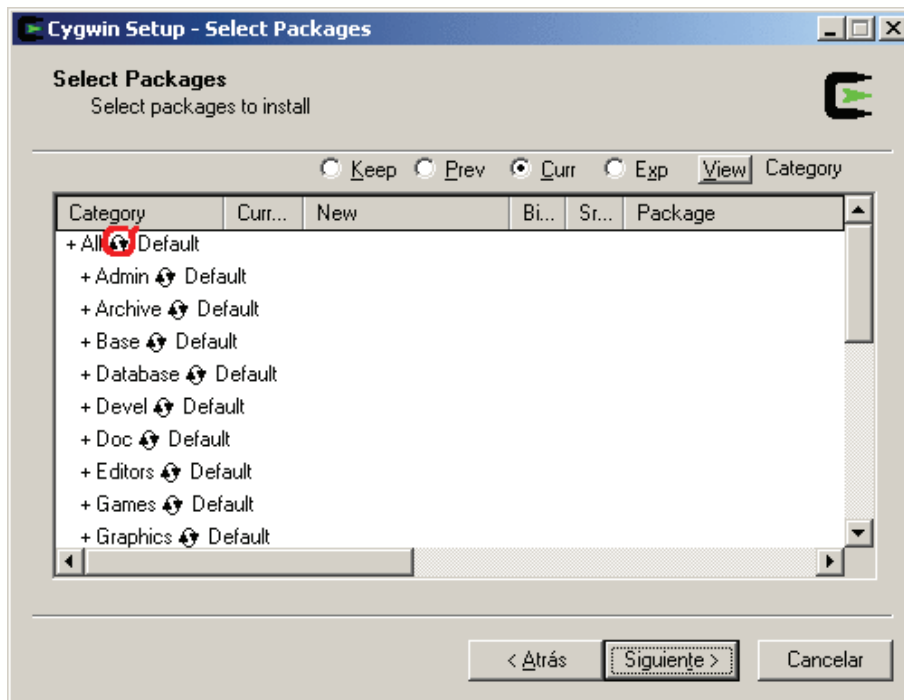


Fig. 22 Selección de componentes a Instalar.

Es en este punto donde los paquetes disponibles representan la posibilidad de llegar a una instalación de hasta 1.5 GB.

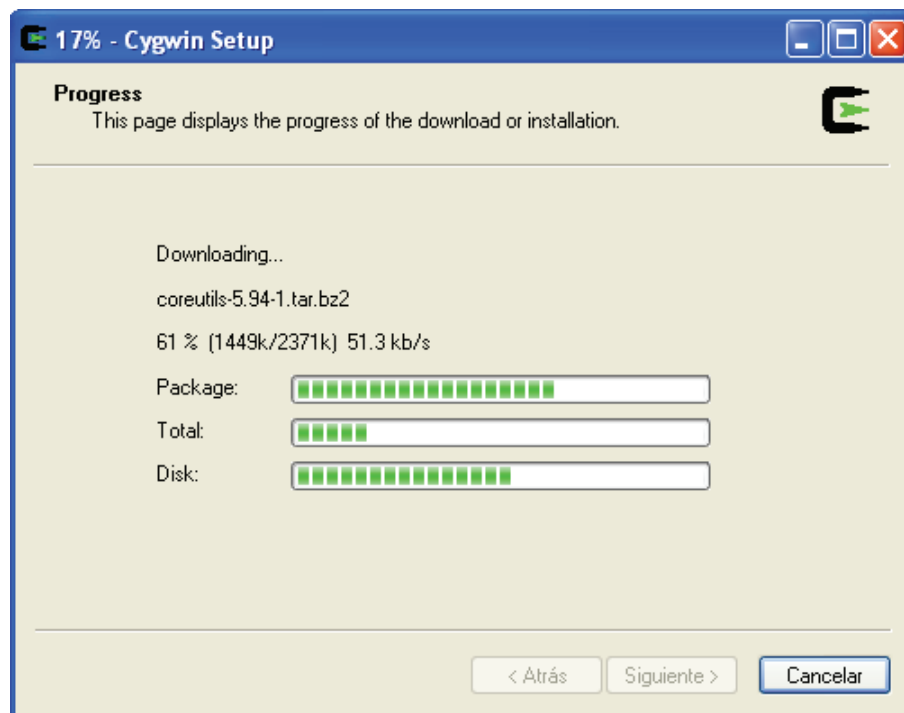


Fig.23 Progreso de descarga o instalación

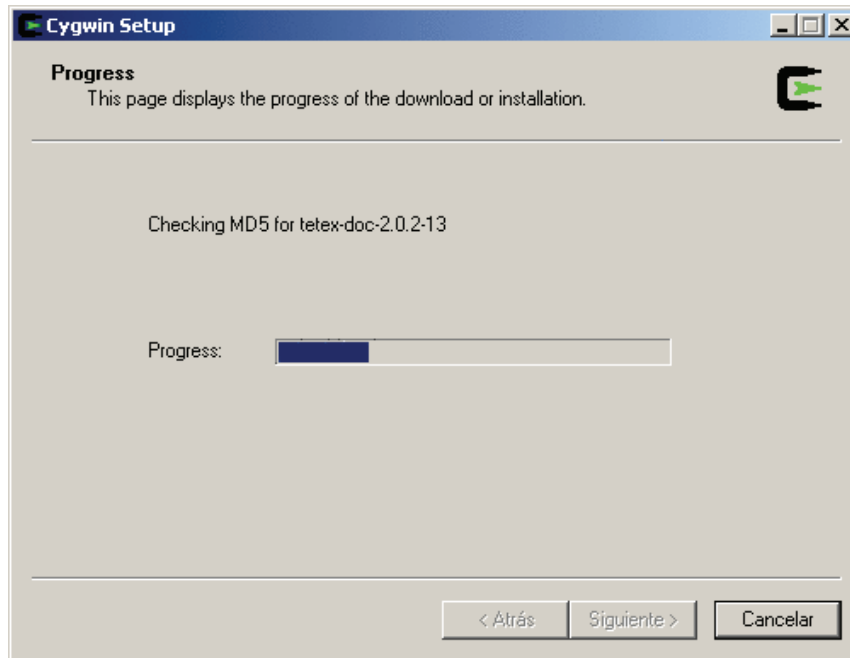


Fig 24 Barra de avance de la etapa de verificación de integridad de componentes.

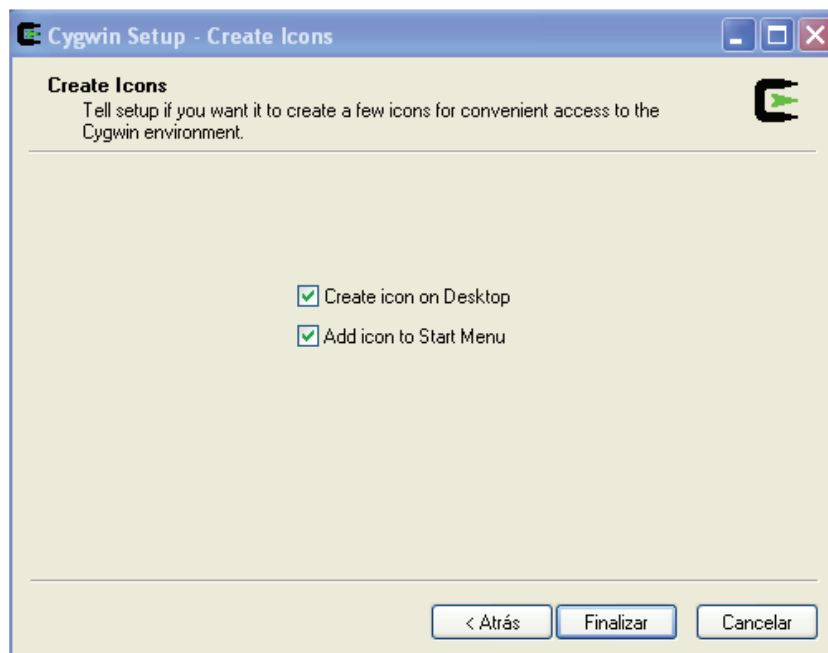


Fig. 25 Opciones adicionales solicitadas por el programa de instalación

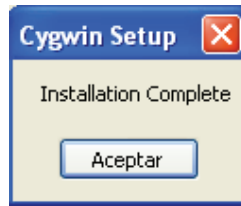


Fig. 26 Instalación completa

Una vez instalado Cygwin se encuentra un icono en el escritorio que lleva a una Terminal similar a la disponible en sistemas UNIX/Linux. Desde esta Terminal se podrán ejecutar comandos UNIX/Linux y de compilación usando comandos típicos de UNIX/Linux tales como gcc.



Fig.27 Acceso directo a Cygwin

4.1.3 Uwin

Uwin es un producto que podría clasificarse como un proyecto de investigación, es decir aún no es un producto comercial y tal vez nunca lo sea; incluso no se encuentra disponible como un producto totalmente terminado. Es por esto que la instalación de esta herramienta es omitida en este apartado.

La primer versión publicada de Uwin es la 4.0 (además sigue siendo la única); esto no significa que nombraron a su primera versión con el numeral 4, sino que las versiones anteriores jamás han sido publicadas y solo se dispone públicamente de esta. Sin embargo a pesar de ir en su versión 4, la ausencia de versiones anteriores que hayan podido ser probadas suficientemente por comunidades de desarrolladores y usuarios, la convierten en una herramienta muy poco confiable y posiblemente vulnerable. De hecho el sitio del proyecto refleja que se trata de un proyecto meramente experimental y no busca convertirse en una plataforma de uso general.

Debido a que el objetivo de esta tesina es probar soluciones viables y con resultados útiles, se considera que es infructuoso experimentar con un producto que todavía se encuentra en fase de investigación, carente de soporte, estabilidad, madurez, etc. Se obtendrían como resultado datos sin valor real en cuanto a su posible aplicación en empresas con servidores en producción, para las cuales significaría un alto riesgo, lujo que muy pocas empresas estarían dispuestas a correr.

4.2 PRUEBAS

Como una prueba de los sistemas a analizar, se eligió la configuración de un servidor NFS completo, con la intención de verificar su integración transparente y funcionamiento básico. A continuación se explica el proceso realizado.

4.2.1 Services for Unix

El objetivo de esta prueba consiste en comprobar que el servidor **NFS** del subsistema para aplicaciones Services for Unix permita el acceso de forma transparente a archivos y directorios ubicados en un servidor; que usuarios y programas traten archivos remotos como si fueran locales independientemente de su plataforma.

Para la prueba a realizarse se cuenta con el escenario siguiente: Se tiene un servidor Windows con SFU que exporta por NFS un directorio llamado: **/win** cuyo sistema de archivos es **NTFS** y tenemos un cliente Linux Fedora Core 5 que monta de manera remota ese directorio como **/mnt/nfs**; no será necesario que nuestro Linux Fedora Core 5 tenga soporte para **NTFS**, sino que simplemente tenga el cliente **NFS**.

La simplificación de dicho escenario se observa de la siguiente manera:

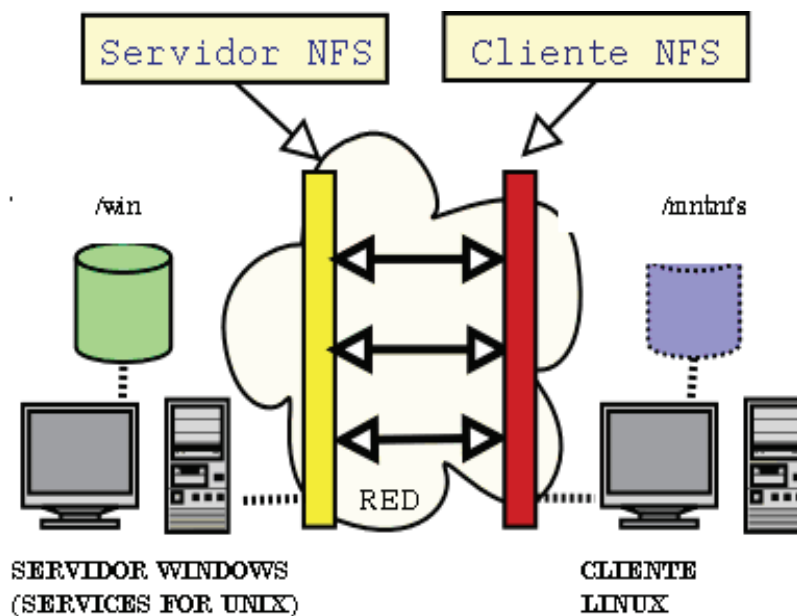


Fig.28 Escenario de pruebas

La computadora remota llamada **pruebas** (Server NFS) pone ciertos directorios de su propio sistema de archivos a disposición de otras computadoras exportándolos hacia la red. Para acceder a los archivos remotos, las computadoras que los utilizarán (clientes **NFS**), hasta

después de encontrarse disponibles por parte del servidor deben montar alguno de los directorios exportados por el servidor en algún punto de su sistema de archivos.

En la siguiente grafica se muestra como el servidor **pruebas** (**Server NFS**), actúa como servidor NFS, ya que exporta a la red el directorio **/win**, y que un cliente NFS llamado **gabriela**, lo monta en **/mntnfs**. Esto significa que para los usuarios de gabriela existe el directorio **/mntnfs**, pero en realidad dichos directorios están ubicados en la maquina **pruebas**.

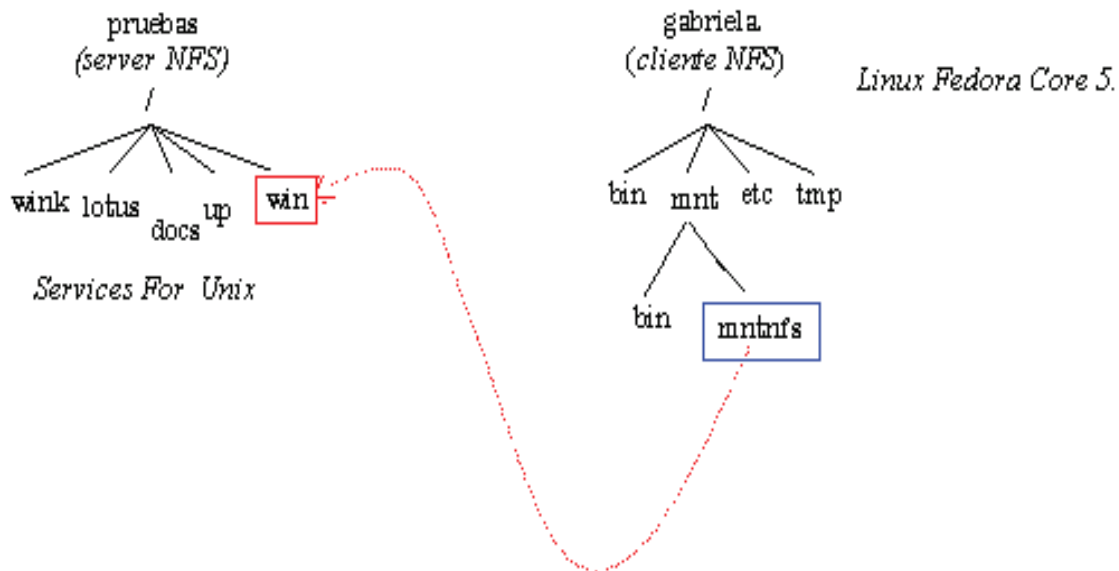


Fig.29 Montaje de directorio

El servidor NFS dispone de herramientas de administración gráficas y de línea de comandos con opciones para configurar el servidor y para el seguimiento (log) de todas las actividades de NFS. El Servidor para NFS de Services for Unix comparte sus recursos utilizando exports NFS.

Windows debe saber a que usuarios permitir los accesos y que tipos de accesos a directorios compartidos. Lo ideal para centralizar los derechos de acceso seria usar un servidor NIS, LDAP o Active Server pero para este caso de derechos de acceso se hizo de la forma más simple, mediante el mapeo de nombres de usuarios y el cual se configura en la consola de Administración de Services for Unix. Este permite el control de accesos basado en las actuales identidades UNIX/Linux y Windows mediante una 'sincronización' de usuarios, con la finalidad de que ambos sistemas, tanto el cliente como el servidor se administren de una forma mucho más sencilla.

El servidor NFS permite compartir directorios y establecer permisos de acceso fácilmente. La solapa "NFS Sharing" es una interfaz gráfica accesible desde el menú de contexto del directorio (haciendo clic con el botón derecho sobre el nombre de la carpeta en el Explorador de

Windows). La utilidad de línea de comandos, `nfsshare`, permite la gestión de recursos compartidos mediante scripting desde shells Windows y UNIX/Linux. Los permisos de acceso NFS se pueden configurar como lectura, escritura y permiten el control de acceso a root para máquinas individuales o grupos de máquinas.

Para el cliente NFS de UNIX/Linux el proceso es totalmente transparente y no requiere software adicional. El acceso a archivos está regido por las identidades de usuario y grupo (UID y GID).

Montaje del Sistema de Archivos NFS en maquina cliente

Una vez que se encuentra el servidor funcionando se procede a acceder al espacio compartido desde una maquina cliente.

```
mount -t<tipo><servidor>: <directorio compartido> <punto de montaje>
```

Así para esta prueba se ejecuto como root, lo siguiente:

```
mount -t nfs 10.43.16.76:/win /mntnfs
```

Donde **10.43.16.76** es la dirección IP de nuestro servidor SFU, el directorio **/win** es el recurso compartido y **/mntnfs** es el directorio de montaje en el cliente.

De esta manera con un simple: **ls /mntnfs** se permite ver que ya se tiene acceso al archivo del servidor.

Es importante recordar, que como en capítulos anteriores se menciona que el servidor NFS soporta los exports NFS desde sistemas de archivo CDFS y NFS únicamente. No están soportados los sistemas FAT y FAT32.

4.2.2 CygWin / Uwin

Cygwin busca la recreación completa de un ambiente UNIX/Linux, en la capa superior al SO Windows. Al ser una recreación completa se busca total y absoluta compatibilidad, por lo que si se ha portado un paquete UNIX/Linux a Cygwin, es seguro que se configurara de la misma manera, se usara de una manera igual y probablemente tendrá un funcionamiento similar. Es por esto que se considera innecesario documentar una configuración de UNIX cuya única diferencia es que se ejecuta con la ayuda de un emulador.

Realizar una prueba similar a la realizada con SFU, pero con Cygwin seria trivial, dado que la mayor dificultad seria 'instalar' el paquete portado (lo cual se realiza ejecutando el asistente y 'palomeando' el componente, para que sea descargado e instalado); el resto sería como usar una maquina UNIX/Linux ordinaria, es decir, para un administrador UNIX/Linux este proceso no requiere un tipo de conocimiento previo extra, lo que es un acierto. Esto no ocurrió con SFU, ya que era configurar e instalar un servicio ajeno totalmente a la plataforma para simular un ambiente o paquete UNIX/Linux, es por ello se documentó con detalle.

4.3 Comparativas entre herramientas

Tanto Microsoft Services for Unix como Cygwin están disponibles de forma gratuita dentro de la red de redes y la diferencia básica entre ambas herramientas es básicamente su estructura interna. En ambos casos se proporciona un entorno UNIX/Linux. Sin embargo Cygwin recurre a una capa de 'emulación' basada en una librería capaz de transformar las llamadas al sistema por la de Win32, mientras que Microsoft Services for Unix cuenta con un entorno denominado Interix con todas las herramientas compiladas directamente para Win32, lo que podría ofrecer un rendimiento superior. Sin embargo Cygwin al ser una recreación completa se obtiene una absoluta compatibilidad y una gran flexibilidad casi a la par de un ambiente UNIX/Linux.

4.4 Conclusiones de Capitulo

Aunque fue posible hacer el experimento exitosamente con SFU, se está limitado a emular ciertos servicios o facilidades disponibles en UNIX/Linux, pero las opciones disponibles son muy limitadas con respecto a un ambiente UNIX real.

Por su parte Cygwin provee, en los paquetes portados, una compatibilidad y similaridad prácticamente perfecta con un ambiente UNIX/Linux, esto aunado a la gran cantidad de utilerías y paquetes disponibles. Aquí no solo se conjuga un ambiente UNIX/Linux muy completo, sino una brecha muy pequeña de aprendizaje para un administrador UNIX/Linux actual.

CAPITULO V
CONCLUSIONES

A continuación se presentan las conclusiones obtenidas a lo largo de este trabajo y algunas perspectivas a futuro.

El propósito del presente proyecto es analizar diferentes opciones validas para la ínter operación entre sistemas UNIX / Windows mediante el uso de subsistemas UNIX, y así explorar una nueva forma de Administración de Servidores; es decir aplicar estas herramientas de integración que originalmente funcionan de manera óptima siempre y cuando se ejecuten en su plataforma nativa, en una plataforma emulada donde la mayor degradación no está en la velocidad de ejecución, sino en que la arquitectura donde se emula tiene filosofías distintas y muy probablemente sea incapaz de recrear por completo un ambiente UNIX/Linux real.

Este estudio tuvo el objeto de conocer, experimentar, evaluar y documentar el funcionamiento de dichos subsistemas de aplicaciones UNIX. Dicho propósito fue alcanzado mediante la investigación del tema, la implantación de dicho ambiente, experimentación del mismo y su evaluación al probarlo.

En la fase de implementación se construyó un entorno UNIX/Linux sobre una plataforma Windows, e implantó un Servidor NFS completo sobre dicha plataforma, prueba simple que se realizó y funcionó de manera satisfactoria.

Se obtuvo una interesante experiencia en cuanto a aprendizaje se refiere, sin embargo para aplicarlo en la realidad se concluye es poco viable dado que lo ofrecido es una emulación, lo cual podría resultar útil en casos experimentales, pero en entornos de producción implica desventajas y riesgos que no se tienen en la plataforma original, como son:

- Pobre Rendimiento
- Hereda el funcionamiento del sistema operativo huésped, así como sus debilidades y limitaciones, que se deben a la arquitectura del mismo.
- No se cuenta con todas las funcionalidades de la plataforma nativa.
- Hereda los problemas de seguridad del sistema operativo huésped.

En síntesis, la investigación resultó satisfactoria en el aspecto académico y de investigación, pero no como una alternativa real en ambientes de producción; además, curiosamente, SFU a pesar de ser un producto de Microsoft cuya filosofía es desarrollar productos altamente simplificados y completos, SFU no se encuentra del todo documentado, ni cuenta con la facilidad de uso típico de la plataforma original; incluso da la impresión de tratarse de un producto en fase de prueba (beta). Por lo tanto no representa una ventaja representativa sobre la arquitectura emulada y tampoco sobre otros métodos de sustitución del ambiente UNIX/Linux. En cuanto a las circunstancias de espacio y configuración necesaria cuenta también con limitaciones inherentes a la arquitectura sobre la plataforma en la cual se encuentra, ya que las hereda de la misma.

Las otras alternativas libres que son Cygwin y Uwin son probablemente algo más complejas y menos intuitivas pero bastante mejor equipadas que su equivalente comercial. Y no es que las aplicaciones propuestas por Microsoft sean escasas, mas bien las contenidas en las herramientas libres son muy abundantes. Además, gran parte de ellas pueden configurarse una vez instaladas por un administrador *nix sin necesidad de tener ningún tipo de conocimiento previo, lo que es un acierto. Sin embargo en ambos casos permanece la desventaja de tener un rendimiento muy

distinto al de la plataforma nativa quedando también reducido a la investigación y casos experimentales.

Probablemente con un rango pequeño de aplicaciones y aplicado en un entorno de bajo riesgo podrían utilizarse estos subsistemas UNIX estudiando previamente las ventajas y desventajas que representa la aplicación en dicha situación particular.

Su implementación solo se sugiere en empresas donde sus aplicaciones actuales estén diseñadas para ejecutarse en el sistema operativo UNIX/Linux y por alguna razón de la empresa, principalmente políticas de la misma, o bien, viabilidad, soporte, costos, o mera facilidad de uso fuera necesario migrar el servidor a plataforma Windows sin posibilidad de conservar algún servidor con la plataforma nativa. En este caso, el implantar SFU o Cygwin sería de bastante utilidad, dado que podría seguir operando y utilizar dichas aplicaciones en el servidor Windows sin necesidad de tener que rediseñarlas.

Se sugiere también su implantación en redes de instituciones académicas para su estudio, probar su funcionalidad, comportamiento; en resumen, para la obtención de experiencia útil que ayude a proponer futuras soluciones, eficientes en rendimiento, seguridad y flexibilidad en ambientes de plataforma mixta.



APENDICE O ANEXOS

CASO DE IMPLANTACION

RADIOSHACK²⁰

INTRODUCCION

RadioShack es uno de los distribuidores de electrónica mas conocidos del mundo, con aproximadamente 5.100 tiendas en todo el mundo y 1.800 puntos de venta franquiciados, a los que la gente se dirige a comprar pilas, juguetes, teléfonos, PCs y muchas otras cosas. Sin embargo, los sistemas de punto de venta, basados en UNIX/Linux y con más de 11 años de antigüedad, han llegado al final de su vida útil y tenían que sustituirse. Después de un costoso proceso de evaluación en el que RadioShack comparó Windows y Linux, la compañía decidió por Microsoft® Windows Server System™ y Windows XP Embedded debido a que la plataforma proporcionaba menores costos a largo plazo, menos riesgos, mejor alineamiento con la estrategia tecnológica a largo plazo, mejor soporte y un mejor aprovechamiento de las capacidades actuales de desarrollo y operaciones de TI (Tecnologías de Información). Esta compañía migró a Windows reduciendo el número de servidores en las tiendas en un 50% y ahorrándose millones de dólares en hardware, software, y costos de administración de sistemas de soporte.

SITUACIÓN

RadioShack es una de las mayores cadenas de distribución de electrónica de consumo, como dispositivos inalámbricos, repuestos de electrónica, pilas, accesorios, y una gran variedad de productos y servicios relacionados con la tecnología. La empresa tiene unas 5.100 tiendas propias y 1.800 en régimen de franquicia en Estados Unidos. Se estima que el 94% de la población de Estados Unidos vive o trabaja menos de cinco minutos de una tienda RadioShack o un distribuidor suyo.

A principios de los 80, cuando empezaba su actividad, RadioShack había sido uno de los principales promotores del cambio que había transformado la industria electrónica. Mas aún: con unos 33.000 empleados, la compañía es un gran consumidor de material electrónico, en concreto, los sistemas informáticos necesarios para mantener la actividad de una empresa, con miles de puntos de venta distribuidos por todo el país. Con el paso de los años, RadioShack ha adoptado distintas plataformas tecnológicas para gestionar sus operaciones de venta de largo alcance y mantener dichas operaciones dentro de su infraestructura corporativa.

A fines de 2003, RadioShack se vió obligada a reexaminar las condiciones de trabajo de la infraestructura tecnológica de sus tiendas para poder soportar un crecimiento sostenido de la actividad comercial. Hasta entonces la compañía había estado utilizando un servidor basado en UNIX en cada una de sus 5.100 tiendas propias para soportar los dispositivos de punto de venta (TPV) y kioscos, sumando unos 22.000 PCs que corrían en Microsoft Windows 95 ò 98. Un segundo servidor en la tienda, con sistema operativo Microsoft Windows 2000 Server, controlaba la conectividad con la sede central de la empresa y el routing básico de la red de la propia tienda, y servía también como estación de trabajo para los empleados.

²⁰ <http://www.microsoft.com/spain/servidores/hechos/casestudies/cs-radioshack.msp>

La infraestructura de TI de la compañía se veía afectada por una serie de problemas de tipo técnico y empresarial, como la obsolescencia de los servidores UNIX, y que hacían cada vez más complicado su soporte. RadioShack ya no podía encontrar un hardware de servidor capaz de ejecutar la versión de SCO UNIX de más de seis años de antigüedad que tenían, y las actualizaciones de software no estaban disponibles para este sistema operativo tan antiguo. Además, la compañía quería consolidar sus dos servidores de las tiendas en uno solo. Con dos plataformas tecnológicas en cada tienda, no solamente se perdía espacio físico, sino que exigía mantener dos grupos de expertos de TI y generaba una carga de trabajo adicional para el grupo de operaciones de TI de la compañía.

La dirección de RadioShack tenía que tomar rápidamente una decisión. La compañía identificó tres posibles alternativas:

- Actualizar a una versión más reciente de SCO UNIX.
- Migrar a una solución basada en código abierto, sobre Linux Red Hat.
- Homogeneizar toda su infraestructura de TI de las tiendas basándose en plataforma Microsoft Windows, incluyendo el software integrado de servidor Microsoft Windows Server System, para los servidores de las tiendas y sistema operativo Windows XP Embedded para los TPV y kioscos de punto de venta.

RadioShack dispuso un sistema de valoración que ponderaba pros y contras de cada opción para diversas categorías de análisis: costos iniciales y a largo plazo, riesgo tecnológico global, nivel de alineamiento con la estrategia tecnológica y objetivos de la compañía de largo plazo, soporte del fabricante, disponibilidad de personal con experiencia en desarrollo de software y operaciones de TI, y tiempo de puesta en marcha de la solución SCO UNIX fue desestimado casi desde el principio, debido a su elevado riesgo empresarial, lo dejaba como alternativas a Windows y Red Hat Linux.

SOLUCIÓN

Una vez terminado el proceso de evaluación, RadioShack llegó a la conclusión de que Windows Server System y Windows XP Embedded ofrecían la alternativa con menor riesgo, más manejable y con mejor relación calidad/precio “Comparamos Windows contra Linux por separado para los servidores TPV, y Windows fue la opción con mejor calidad/precio en ambos casos”, dijo Ron Cook vicepresidente y Director de Tecnología de RadioShack “Windows nos proporciona una plataforma integrada, fácil de administrar que se puede ampliar desde el BackOffice hasta la línea de cajas. Si hubiésemos optado por Linux en cualquiera de las áreas, los costos finales y la complejidad serían muy superiores”.

La nueva infraestructura de punto de venta, entró en funcionamiento en septiembre del 2005, incluye:

- Microsoft Windows Server™ 2003 Ed. Enterprise (el sistema operativo base de Windows Server System), al que se actualizarían los actuales servidores Windows 2000 Server de cada tienda.
- Microsoft Windows XP Embedded que sustituye a los TPV basados en Windows 95 y Windows 98, y a los kioscos de las tiendas.

-
- Microsoft Systems Management Server 2003, que controla la distribución de actualizaciones de software para los servidores, TPVs y kioscos en toda la red.
 - Microsoft Services for Unix, versión 3.5, permitió a RadioShack adaptar sus aplicaciones de punto de venta basadas en UNIX para poder correr en servidores basados en UNIX.

SUPERVIVENCIA DE LA APLICACIÓN UNIX

Una de las razones principales por la que RadioShack optó por Windows Server System era Services for Unix 3.5, que ofrecía una posibilidad de alargar la vida útil de su aplicación de punto de venta, basada en UNIX.

RadioShack solicitó la colaboración de Microsoft Services for Unix para dar soporte a los programadores de RadioShack que modificaron las aplicaciones de punto de venta para poder funcionar bajo Services for Unix 3.0, un esfuerzo que finalizó en febrero de 2005. “Services for Unix nos dio vía para migrar nuestra aplicación de punto de venta hacia una plataforma mas manejable y económica sin tener que dedicar un montón de tiempo y energía en desarrollar un nuevo software”. Comentó Cook.

La aplicación modificada fue ya distribuida por más de 1.200 tiendas. Los primeros resultados fueron enormemente positivos. La compañía pensó en finalizar el despliegue de Windows Server 2003, Services for Unix y la nueva aplicación POS en las tiendas de la cadena para finales de septiembre de 2005. Cuando se completó el proceso de implantación RadioShack vio reducido el número de servidores en cada establecimiento de dos a uno y elimino por completo la necesidad de mantener y dar soporte a sistemas obsoletos basados en UNIX. Toda la funcionalidad de BackOffice se ejecutó en el nuevo sistema basado en Windows, incluyendo la aplicación de punto de venta y el PC de gestión de mostrador que era necesario en cada tienda.

“Con la infraestructura antigua se tenían dos servidores en cada establecimiento para gestionar las funciones de la tienda y para enviar datos de punto de venta y de otro tipo a la oficina central” dice Cook. “Era complicado y costoso mantener, tanto en términos de mantenimiento de hardware como gestión del sistema. Con Windows Server System ya se puede soportar cada establecimiento con un solo servidor, reduciendo a la mitad el numero de sistemas que tienen que administrarse y mantenerse”.

BENEFICIOS

Según Cook la solución basada en Windows elegida por RadioShack fue fácil de administrar y económico que la basada en Linux, y le ofrece a la compañía una sólida base tecnológica para dar soporte a sus operaciones de distribución y venta en los próximos años. Entre los beneficios que la compañía obtuvo eligiendo Windows aparecen en un menor costo total de propiedad, un procedimiento mas sencillo de administrar y distribuir las actualizaciones de software.

“Windows ofrecía una serie de ventajas generales significativamente superiores a las que ofrecía Linux, aun contando con el bajo costo inicial de las licencias de Linux”, comentó Cook. Windows Server System es un paquete mucho mas integrado y completo. Si hubiésemos elegido Linux nos habrían surgido costos adicionales, ya que hubieran tenido que conseguir el nivel de conocimiento necesario para el desarrollo y gestión del sistema y desde luego la oferta de productos y tecnologías de terceros seria mucho mas limitada, incluyendo aplicaciones y drivers.”

NUEVAS OPCIONES PARA UN CRECIMIENTO FUTURO

De acuerdo con Cook, RadioShack ha elegido Windows Server System y Windows XP Embedded no solo por sus beneficios económicos inmediatos, sino también por sus posibilidades a largo plazo. El cree que es importante poder disponer de un fabricante que trabajara para integrar sus soluciones en ciertas tecnologías emergentes, esenciales en el sector de la distribución, como RFID (*Radio Frequency Identification*).- minúsculos microchips que pueden responder a peticiones basadas en emisión de radio transmitiendo códigos de identificación únicos.

“Microsoft mantiene un elevado grado de compromiso con su estrategia para el sector de la distribución, lo que supone se puede contar con que la plataforma Windows va a responder a nuestras necesidades y objetivos de negocio”.Finalizo Cook.

REFERENCIAS

- [1] Lonnie HARVEL, David WEBB, Steven FLYNN, and Todd WHITEHURST
The UNIX and Windows 2000 Handbook: Planning, Integration and Administration
(Paperback - May 4, 2000)
Prentice Hall PTR; 1st edition (May 4, 2000)
- [2] Gene HENRIKSEN *Windows and UNIX Integration*
Macmillan Technical Pub (May 1998)
- [3] MACMILLAN *Windows NT & UNIX Integration (Hardcover)*
MacMillan General Reference
MacMillan Publishing Company (July 1, 1998)
- [4] Steve BURNETT, David GUNTER, Lola GUNTER, Wendy RINALDI
Windows 2000 & UNIX Integration Guide
- [5] MICROSOFT *Services for Unix. 3.0*
July 12, 2002
- [6] Jeffrey R.SHAPIRO, Jim BOYCE, Marcin POLICHT, Brian PATTERSON,
Scott, LEATHERS
Windows Server 2003 Bible (Paperback)
Wiley (December 1, 2002)
- [7] *Windows and UNIX interoperability* / Sitio Oficial de Microsoft
Microsoft Corporation
<http://www.microsoft.com/latam/technet/articulos/200007/art07/default.asp>
2007-05-09
- [8] *Windows Services for UNIX* / Sitio Oficial de Microsoft
Microsoft Corporation
<http://www.microsoft.com/windows/sfu>
2007-05-09
- [9] *Cygwin* / sitio oficial de Cygwin
Red Hat, Inc
<http://www.cygwin.com/>
2006-03-12
- [10] *Uwin overview* / Sitio de AT&T
AT&T
<http://www.research.att.com/sw/tools/uwin/>
2006-02-27
- [11] *Sfu 3.0* / página de desarrollo Web
<http://www.desarrolloweb.com/articulos/814.php>
2002-06.13

-
- [12] ***Aprendiendo a aprender linux / Structio***
http://structio.sourceforge.net/guias/AA_Linux_colegio/administracion-de-usuarios.html
2003-02-22
- [13] ***La alternativa UNIX / Linux a los servidores Microsoft***
<http://es.tldp.org/Informes/conf-MigraNT2GNU/doc-migrar-nt-linux-html/tema02.html>
2007-05-09
- [14] ***Conference and Expo / Sitio Linux world***
Beth WICKENHISER
<http://www.linuxworldexpo.com/live/12/media//news/CC713169>
2004-01-14
- [15] ***Windows 2003 vs. Linux file server***
<http://www.kegel.com/nt-linux-benchmarks.html>
2003-10-14
- [16] ***NT vs Linux server Comparisions/ It Week Labs***
Dan KEGEL
<http://www.itweek.co.uk/itweek/news/2085218/samba-extends-lead-win-2003>
2003-10-14
- [17] ***Virtual PC / Sitio oficial Microsoft***
Microsoft
<http://www.microsoft.com/spain/windows/virtualpc/default.aspx>
2004-18-09
- [18] ***Winfs / Wikipedia, Enciclopedia***
Wikipedia Project
<http://es.wikipedia.org/wiki/WinFS>
2007-03-09
- [19] ***Desarrollo / Lycos***
<http://usuarios.lycos.es/hv1102/desarrollo.html>
2006-15-02
- [20] ***The Xen virtual machine monitor / Sitio Oficial Xen***
University of Cambridge Computer Laboratory
<http://www.cl.cam.ac.uk/Research/SRG/netos/xen/index.html>
2007-05-08
- [21] ***General Public Licence / Proyecto GPL***
Webmasters GPL
<http://www.gnu.org/licenses/licenses.es.html>
2004-03-07

[22] *Casos de Éxito* / Sitio oficial Microsoft

Radioshack

[http://www.microsoft.com/spain/servidores/hechos/
casestudies/cs-radioshack.aspx](http://www.microsoft.com/spain/servidores/hechos/casestudies/cs-radioshack.aspx)

2007-005-09