

Universidad de las Ciencias Informáticas

Facultad 6



**Título: Distribución de Linux específica para
Bioinformática**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores: Mabelis Padrón Fernández
Giralte Bello Ferrer

Tutores: Ing. Héctor Rodríguez Figueredo
Ing. Noel Moreno Lemus
Ing. Orlando Martínez

Junio 2007

"Si queremos asegurar algún logro social, como por ejemplo la salud pública o el medio ambiente, siempre necesitaremos el software libre"

Richard Stallman

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días, del mes de _____ del año _____.

Autores:

Mabelis Padrón Fernández

Giraltne Bello Ferrer

Tutores:

Ing. Héctor Rodríguez Figueredo

Ing. Noel Moreno Lemus

Agradecimientos

Inicialmente queremos agradecer a la Revolución, por hacer de la educación cubana un derecho de todos y contribuir cada día con su desarrollo; permitiendo y apoyando la creación de centros como el nuestro.

Nuestros más sinceros agradecimientos son para el Comandante en Jefe Fidel Castro, por haber concebido una universidad como la UCI y hacerla extensiva a todo el país. Agradecemos enormemente a nuestra familia por apoyarnos y comprendernos en todo momento.

Les damos las gracias a nuestros tutores por el apoyo incondicional que nos han brindado durante la realización de esta tesis, así como a las facultades 6 y 10; en especial al grupo de trabajo de la facultad 10 por el tiempo que nos dedicaron y la ayuda que nos brindaron.

Les damos las gracias a todos los amigos y compañeros que estuvieron a nuestro lado y confiaron en nuestra capacidad para lograr nuestro objetivo.

Y para terminar, queremos agradecer al amor y a la vida, que hicieron posible que este día se hiciera realidad.

Sinceramente, muchas gracias a todos!

Dedicatoria

A todas aquellas personas que quieran aprovechar las ventajas que ofrece GNU/Linux para el mejor desempeño de su trabajo.

Resumen

El siguiente trabajo describe el desarrollo de un LiveCD basado en una distribución de GNU/Linux; compuesto por aplicaciones puramente bioinformáticas. El objetivo que se persigue con el mismo es brindar una nueva distribución para ser utilizada por los centros que se dediquen a las investigaciones biotecnológicas y que utilizan la Bioinformática como herramienta para el procesamiento de datos y el análisis de la información relacionada con las investigaciones en biotecnología.

La meta distribución tomada como punto de partida para el desarrollo de la misma es Nova, debido a las múltiples ventajas que ofrece y con el propósito de aprovechar el trabajo realizado por un equipo de desarrollo de la Universidad de Ciencias Informáticas.

Como resultado final se espera obtener un LiveCD que incluya aplicaciones para el trabajo en las diferentes ramas de la Bioinformática, como son: la Biología Molecular, la Química Teórica y la Biología de Sistema.

Palabras claves:

- Bioinformática.
- Distribución.
- LiveCD.
- Repositorio.

Indice

Agradecimientos.....	I
Dedicatoria	II
Resumen	III
Introducción.....	1
Capítulo 1: Fundamentación Teórica	5
Desarrollo de la Bioinformática	5
¿Por qué nos surge la necesidad de una distribución de GNU/Linux específica para Bioinformática?	9
Estado del Arte	10
BioBrew	11
BioKnoppix	12
SaluX.....	13
DNA Linux	15
Linux Hélice.....	16
Principales distribuciones de GNU/Linux	18
Debian	18
Mandriva.....	20
Ubuntu.....	21
Suse	24
Red Hat	25
Gentoo.....	27
Distribución a utilizar como punto de partida	30
Capítulo 2: Descripción del Proceso de Desarrollo de una Distribución de GNU/Linux.....	33
Metodología general para la creación de una distribución de GNU/Linux.	33
Ingeniería del Software Libre.	34
Fases de la Ingeniería de Software Libre:.....	35
LiveCD.....	38
Objetivos para crear un LiveCD.	38
Ventajas de la creación de un LiveCD	39

Requerimientos mínimos para crear un LiveCD.....	39
Elaboración de un LiveCD.....	40
Capítulo 3: Desarrollo e Implementación de una Distribución de GNU/Linux	51
Arquitectura de Nova BioLive.....	51
Kernel.....	52
Base de usuario GNU	52
Base de usuario Gentoo.....	52
Nova CLI	52
Nova GUI.....	53
Nova BioLive Userland.....	53
¿Cómo se creó el LiveCD de Nova BioLive?	53
Pasos para instalar Nova	53
Preparación de la imagen.....	57
Instalación de las aplicaciones.....	60
Creación del LiveCD	60
Características propias de Nova BioLive	67
Paquetes de Nova BioLive	67
Biología Molecular.....	67
Química Teórica.....	69
Biología de Sistemas.....	70
¿Por qué se escogieron estas aplicaciones?	71
Calidad	73
Conclusiones.....	75
Referencias Bibliográficas.....	76
Anexos	81
Anexo 1:	81
Anexo 2:	82
Glosario de Términos.....	83

Introducción

La Sociedad Cubana está cambiando rápidamente, los avances tecnológicos se suceden unos a otros a una velocidad increíble, la informatización cobra nuevos bríos bajo la égida de una Revolución cuya mayor riqueza es el capital humano que posee. Cada día se compran nuevas computadoras y se instalan en escuelas, empresas y centros de trabajo, sin embargo, todas traen instalado Windows, sistema que profundiza la dependencia tecnológica y drena los recursos de las economías de muchos países. Una alternativa esta cerca: GNU/Linux ha ganado el lugar que le corresponde entre quienes visualizan la magnitud del problema y suman sus esfuerzos en la búsqueda de la solución, ya es un hecho que la migración a este sistema libre es una necesidad vital si se quiere desarrollar la Industria del Software.

Al ser GNU/Linux un sistema nuevo en el país, el personal y los profesionales de la Informática en general no se encuentran preparados para afrontar una migración, se necesita un amplio proceso de divulgación y convencimiento, que ayude a enfrentar el proceso de resistencia al cambio que siempre aparece, debido a los costos psicológicos que tiene para el individuo, el grupo y la organización. Posteriormente sería indispensable un trabajo de capacitación para aprender el funcionamiento del nuevo sistema y sus particularidades.

Hoy día hasta los laboratorios más pequeños son capaces de generar una gran cantidad de datos biológicos. El procesamiento e interpretación de esta información resulta imposible sin el uso de herramientas computacionales. La comunidad científica a nivel mundial se ha dado a la tarea de desarrollar disímiles aplicaciones informáticas que pueden ser aplicadas a infinidad de problemas de diferente naturaleza, dentro de las ciencias biológicas.

El vertiginoso desarrollo alcanzado en las ciencias biológicas y las ciencias de la computación, hace que esta sea un área de trabajo en constante cambio. Resulta, por tanto, sumamente importante la socialización del conocimiento generado, para que la comunidad científica sea capaz de utilizarlo y mejorarlo cada vez más.

El Software Libre se ha convertido en una alternativa viable para el desarrollo de la Bioinformática; rama del saber donde se agrupan las ciencias de la vida y las ciencias de la computación en aras de resolver problemas de interés biológicos, mediante el uso de herramientas computacionales.

La Bioinformática, impulsada por la incógnita del genoma humano y la promesa de una nueva era en la cual la investigación genómica puede ayudar dramáticamente a mejorar la condición y calidad de vida humana, se torna cada día más importante para el desarrollo científico a nivel mundial. Esta ciencia se ha adentrado de forma vertiginosa en el campo de la súper computación, debido esencialmente a la necesidad de grandes recursos computacionales para resolver sus problemas. Por tal motivo es muy común entre los científicos que trabajan estas temáticas el uso de clusters de computadoras o infraestructuras distribuidas (Tecnologías Grid de Computación). Por lo general estas infraestructuras están montadas sobre arquitecturas libres, esencialmente debido a la flexibilidad que el Software Libre proporciona, sobre la rigidez del software propietario.

GNU/Linux es el sistema operativo que se ha usado como plataforma para el desarrollo de la Bioinformática y esto se debe a su excelente rendimiento en el procesamiento de datos, al aprovechamiento óptimo de los recursos de hardware y fundamentalmente por ser distribuido en forma gratuita bajo la filosofía de código abierto.

El problema radica básicamente en que no existe una herramienta capaz de integrar todas las aplicaciones necesarias para el trabajo en las diversas áreas vinculadas a la Bioinformática.

En la actualidad existen grupos encargados de personalizar distribuciones de GNU/Linux para el campo específico de la Bioinformática. Estas metadistribuciones ofrecen amplios beneficios a quienes se ven involucrados directamente en el tema de la biología y la salud.

Hasta el momento no se conocen antecedentes de trabajos similares en Cuba, aunque sí en el extranjero. No obstante estos trabajos relacionados no satisfacen todos los requerimientos necesarios para cumplir con los objetivos que se ha propuesto el siguiente trabajo, aunque las ideas básicas de éstos, han servido como punto de partida para el mismo.

El presente trabajo tiene novedad científica y práctica ya que como resultado se obtendrá un producto completamente funcional con aplicación inmediata en los referidos centros.

El objeto de estudio irá encaminado a la personalización de una distribución de GNU/Linux específica para Bioinformática.

Los resultados obtenidos serán de gran utilidad para Centros de Estudios o Centros de Investigaciones en Biotecnología o en general donde se utilice la Bioinformática como herramienta para el desarrollo de aplicaciones o investigaciones asociadas, siendo este su campo de acción fundamental.

Como principal aporte práctico del trabajo se obtendrá una distribución de GNU/Linux personalizada para la Bioinformática. Se distribuirá en un LiveCD lo cuál no obliga al usuario a tener GNU/Linux instalado en su computadora y todas las funcionalidades están incluidas en la distribución.

Objetivo General:

El objetivo general es desarrollar una distribución de GNU/Linux para Bioinformática. Para conseguirlo se pretende partir de objetivos específicos como:

Objetivos Específicos:

- Identificación de la problemática.
- Desarrollo de una metodología para la personalización de distribuciones de GNU/Linux.
- Desarrollo de una distribución de GNU/Linux personalizada para la Bioinformática.
- Desarrollo de una distribución en formato LiveCD.

Para el cumplimiento total de estos objetivos ha sido fundamental la realización de diversas tareas como:

- Visitas a los centros del polo para ver las necesidades y requerimientos que tienen.
- Revisión en Internet de antecedentes de trabajos similares.
- Trabajo conjunto con el grupo de Software Libre de la Facultad 10 para ver las experiencias obtenidas y tomarlas como punto de partida.

- Estudio de las principales distribuciones de GNU/Linux que existen para seleccionar una a partir de la cual desarrollar la nuestra.
- Confección de una metodología para la creación de una distribución de GNU/Linux.
- Estudio de aplicaciones bioinformáticas de Software Libre más utilizadas por los profesionales de esta rama a nivel mundial.
- Selección de aplicaciones bioinformáticas de Software Libre a incluir en el LiveCD.
- Remasterización de Nova.

El presente trabajo consta de tres capítulos formados por epígrafes y subepígrafes, encargados de dar una panorámica sobre la aplicación que se prevé desarrollar.

El capítulo uno recibe el nombre de Fundamentación Teórica de una distribución de GNU/Linux específica para Bioinformática. El mismo contiene la parte teórica de la tesis y consta de cinco epígrafes cuyo contenido brinda una reseña a cerca del desarrollo de la Bioinformática en el mundo y su impacto en Cuba. Además expone los adelantos del Software Libre a nivel mundial.

El capítulo dos tiene el título de Descripción del proceso de desarrollo de una distribución de GNU/Linux. En él van a ir incluido todo lo concerniente a la realización de una distribución de GNU/Linux en general. Consta de cuatro epígrafes relacionados con la confección de una metodología, la creación de un LiveCD y lo concerniente al estado actual de la Ingeniería del Software Libre.

El tercer capítulo se titula Desarrollo de la Distribución y va a contener las características propias de la distribución obtenida.

Capítulo 1: Fundamentación Teórica

Introducción

Este capítulo recibe el nombre de Fundamentación Teórica de una distribución de GNU/Linux específica para Bioinformática. En él se describe brevemente el desarrollo de la Bioinformática en el mundo y su impacto en Cuba. También ofrece una panorámica de los adelantos del Software Libre a nivel mundial, haciendo especial énfasis en las distribuciones más importantes de este Sistema Operativo. Además ofrece un estudio a cerca de las distribuciones de Linux específicas para Bioinformática que existen en el mundo.

Desarrollo de la Bioinformática

En la década de los años 50, con el descubrimiento de la estructura de doble hélice del ADN por los científicos Francis Crick y James Watson, comenzó una nueva época en el desarrollo de la biología molecular. A mediados de los años 90, la prensa internacional difundió la noticia de la publicación del genoma humano, abriendo definitivamente el paso a la era Genómica.

La disponibilidad de genomas completos, el volumen de información ubicado actualmente en las bases de datos públicas y los ambiciosos proyectos masivos de estudio sobre la interacción entre proteínas ha generado un cambio de paradigma. El enfoque clásico, que consistía en conocer una determinada función y buscar el gen responsable, se transformó y creó un nuevo escenario donde se dispone de un importante número de genes desconocidos a los que es necesario asignar una función. Este nuevo momento dio lugar al desarrollo de la Bioinformática.

Existen múltiples definiciones sobre la Bioinformática. Una de las más completas es aquella que la refiere como una disciplina científica que se interesa por todos los aspectos relacionados con la adquisición, almacenamiento, procesamiento, distribución, análisis e interpretación de la información biológica, mediante la aplicación de técnicas y herramientas propias de las matemáticas, la biología y la informática, con el propósito de comprender el significado biológico de una gran variedad de datos.

En la década de los años 80, Cuba comenzó la introducción y el desarrollo de las técnicas modernas de la biotecnología, justo en el momento en que este campo iniciaba su progreso en el resto del mundo. Se crearon varios centros como es el caso de BIOINFO, el CIM y se integró el Polo Científico del Oeste de Ciudad de La Habana (CIGB), que situó al país en una posición competitiva incluso con respecto a los países desarrollados.

Han transcurrido dos décadas desde aquel entonces; la biotecnología se ha establecido, y lo que inicialmente fue una esperanza es ahora una realidad científica, productiva y económica para Cuba y otros países, principalmente los desarrollados.

Durante estas dos décadas, ha ocurrido un desarrollo sin precedentes en la tecnología del análisis y manipulación de los sistemas biológicos. Paralelamente, se han modificado las concepciones para la investigación en este campo. Los adelantos científicos alcanzados hacen posible que hoy se pueda abordar, con una eficiencia antes inimaginable, la identificación, el análisis, así como la determinación de la actividad biológica y la síntesis de las moléculas biológicas en el orden de los miles, y todo ello simultáneamente.

Los progresos que Cuba ha logrado en el área de la salud han sido protagonizados por el desarrollo alcanzado por la Bioinformática. La sola idea de enfrentar proyectos que aspiren a desarrollar medicamentos y otros productos novedosos con una fuerte posición de patente y en un tiempo relativamente breve, trae implícito la presencia de la Bioinformática. Se hace evidente que si queremos crecer en esta rama, debemos impulsar en nuestro país el avance de la Bioinformática, teniendo en cuenta, fundamentalmente, que una demora de esta actualización tecnológica repercutirá en la eficiencia y la competitividad biotecnológica nacional.

Para este fin, se hizo necesario acometer de inmediato diferentes tareas entre ellas:

- Creación de un grupo de coordinación, que comenzó a laborar desde febrero del 2002 y que se encuentra integrado por las siguientes instituciones:

Polo Científico al Oeste de la capital.

Ministerio de Educación Superior (MES).

Universidad de la Habana (UH).

Ministerio de Ciencia Tecnología y Medio Ambiente (CITMA).

Centro de Cibernética Aplicada a la Medicina (CECAM), incorporado en julio del 2002.

Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE).

- Formación de recursos humanos.
- Creación de centros de Bioinformática.
- Desarrollo de clusters (agrupamiento) de computadoras.
- Desarrollo de la neuroinformática: una fortaleza cubana.
- Realización de congresos y eventos sobre la especialidad. [1]

En nuestro país se ha llevado a cabo un intenso proceso inversionista y de formación de personal, que permite disponer en la actualidad de un complejo e integrado sistema de investigación-producción en la esfera de la Biotecnología aplicada a diferentes ramas de la sociedad.

Un paso importante en el desarrollo de la Biología Computacional en el país, lo constituyó la creación del Centro Nacional de Bioinformática en octubre del 2001. BIOINFO fue creado para ser una institución no lucrativa y como un instituto de investigación del Ministerio de Ciencias, Tecnología y Medio Ambiente de Cuba. Su misión es desarrollar actividades de investigación científica, entrenamiento de los recursos humanos y la prestación de servicios especializados a la comunidad científica en los campos de la computación de alto rendimiento y en el procesamiento de secuencias biológicas.

Uno de sus objetivos principales es convertirse en un centro de referencia de Bioinformática en Cuba. Para lograr esto han concebido un plan ambicioso incluyendo proyectos de investigación con la cooperación de los laboratorios nacionales e internacionales, el entrenamiento especializado y la creación de una plataforma de servicios bioinformáticos que se puede acceder desde Internet, que incluye sistemas

de recuperación de secuencias biológicas, alto rendimiento computacional, sistemas de predicción, entre otros. [2]

En la rama de la Bioinformática también hace sus principales aportes el CIM (Centro de Inmunología Molecular). Este centro nace el 5 de Diciembre de 1994 en el oeste de la Habana y tiene como misión obtener y producir nuevos biofármacos destinados al tratamiento del cáncer y otras enfermedades crónicas no transmisibles e introducirlos en la Salud Pública cubana. Hacer la actividad científica y productiva económicamente sostenible y realizar aportes importantes a la economía del país.

En este centro laboran cerca de 400 trabajadores, en su mayoría científicos e ingenieros de forma multidisciplinaria. Este personal está organizado administrativamente en tres áreas principales: Investigación-Desarrollo, Producción y Aseguramiento de la Calidad.

Actualmente el CIM fabrica productos biofarmacéuticos, tales como: un anticuerpo monoclonal anti CD3 para el tratamiento de pacientes con rechazo del trasplante de órganos, Eritropoyetina humana recombinante para el tratamiento de la anemia, Factor estimulante de Colonias granulocíticas para el tratamiento de la Neutropenia, un anticuerpo monoclonal “humanizado” que reconoce el receptor del Factor de Crecimiento Epidérmico para el tratamiento del cáncer, así como otros anticuerpos para el estudio in vivo por inmunogammagrafía de tumores en diferentes localizaciones. [3]

En este contexto se destaca una institución de desarrollo dinámico, mencionada anteriormente: el Centro de Ingeniería Genética y Biotecnología de Cuba. El CIGB nos ha permitido alcanzar un alto nivel en la investigación, desarrollo, producción y comercialización de productos biológicos obtenidos a través de los métodos de la biotecnología moderna.

En estos momentos cuenta con la mayor potencia de computo paralelo, cluster o grupo de computadoras personales conectadas en serie para resolver tareas complejas, en el país. En este importante centro se llevan a cabo diversas investigaciones en el área de la biología computacional y se coordinan a la vez los esfuerzos bioinformáticos de las instituciones del Polo Científico del Oeste de La Habana.

En este centro se llevan a cabo actividades docentes y de formación de especialistas en Bioinformática. Como parte de estas tareas de formación el CIGB, nodo nacional cubano de la red EMBnet ha contribuido

de forma sistemática al desarrollo de talleres internacionales de Bioinformática que han contado con la participación de especialistas de primer nivel y alumnos de Cuba y otros países latinoamericanos.

Los temas fundamentales de trabajo están enfocados a las enfermedades infecciosas: cáncer, inflamación, auto inmunidad, cicatrización y enfermedades cardiovasculares. [2]

Un principio básico de los científicos de nuestro país es socializar el conocimiento con el propósito de que la comunidad científica sea capaz de utilizarlo y mejorarlo cada vez más. Con este fin y aprovechando las múltiples ventajas que ofrece, se ha establecido GNU/Linux como plataforma oficial en estos centros.

Cuba, atendiendo al creciente auge que ha tenido el Software Libre a nivel mundial, apoya el desarrollo del mismo en el área de la Bioinformática y se prepara para una migración a este sistema libre.

¿Por qué nos surge la necesidad de una distribución de GNU/Linux específica para Bioinformática?

En los últimos años la Informática ha tenido un desarrollo vertiginoso en Cuba. Su campo de acción ha ido creciendo a la vez que nuestra sociedad ha ido alcanzando una cultura adecuada a las necesidades y a los avances tecnológicos que se suceden a diario en el mundo entero. Hoy nuestro país enfrasca su trabajo a informatizar los sectores más fuertes, tal es el caso de la educación y la salud.

Sin embargo la tarea no es tan fácil. Conjunto con el desarrollo y la aplicación de la informática, se ha planteado la necesidad de sustentar el accionar informático en el Software Libre. Esto se debe en gran medida a la dependencia tecnológica que implica el Software Privativo y los obstáculos que engloba; sobre todo para los países del tercer mundo quienes se ven limitados por las costosas patentes. Otra ventaja que motiva la migración al Software Libre es la posibilidad que este ofrece de compartir los conocimientos alcanzados y los resultados obtenidos en aras de otros más novedosos. Esta característica se acopla muy bien al campo de la Bioinformática, donde ya se dan los primeros pasos en este aspecto.

La Bioinformática como herramienta fundamental de trabajo relacionado con los avances científicos, se ha ido adentrando en el campo de la súper computación y hoy muestra grandes avances que recaen

fundamentalmente en su utilización como herramienta para el procesamiento de datos y el análisis de la información relacionada con las investigaciones en biotecnología.

GNU/Linux es el sistema operativo que se ha utilizado como plataforma para el desarrollo de la Bioinformática en el mundo entero y Cuba ha hecho suya esa tradición teniendo en cuenta su excelente rendimiento en el procesamiento de datos, al aprovechamiento óptimo de los recursos de hardware y fundamentalmente el hecho de ser distribuido de forma gratuita bajo la filosofía de código abierto. Además en la Bioinformática actual es muy frecuente el uso de clusters de computadoras o infraestructuras distribuidas (Tecnologías Grid de Computación). Las cuales por lo general están basadas sobre arquitecturas libres, esencialmente debido a la flexibilidad que el Software Libre proporciona, sobre la rigidez del software propietario. El Centro de Ingeniería Genética y Biotecnología, puntero del desarrollo de la Bioinformática en Cuba, dispone de la mayor cantidad de clusters existentes en nuestro país.

Con el propósito de aliviar el trabajo de científicos y biólogos se han conformado grupos de desarrollo que han creado diversas distribuciones específicas para la Bioinformática. Como resultado de su trabajo han surgido distribuciones como BioBrew, BioKnoppix, DNA-Linux, Linux Hélice y Salux.

Estado del Arte

La investigación biológica es bastante heterogénea, su campo de acción es variado y el tipo de análisis posible puede llegar a ser casi infinito. Para cada una de estas tareas se han ido conformando múltiples herramientas con el objetivo de agilizar el proceso de desarrollo de las mismas. Sin embargo estas herramientas no cubren todas las necesidades existentes en el mundo de la biología, por lo que se ha hecho inevitable su modificación en aras de alcanzar los resultados propuestos. El Software Libre nos brinda la posibilidad de partiendo de una distribución existente crear otras novedosas y eficientes. Actualmente las distribuciones más usadas en el campo de la bioinformática son BioBrew, BioKnoppix, DNALinux, Linux Hélice y Salux. A continuación una descripción breve de sus principales características.

BioBrew

Fue construida por Glen Otero en noviembre del 2003. Es una colección de aplicaciones de open-source, la cual fue realizada para la superación de los biólogos y para los científicos de la vida.

Las principales características de BioBrew son:

- Es una distribución de GNU/Linux libre.
- Está basada en RED HAT y optimizada para su uso en clusters.
- Posee una multitud de paquetes para bioinformáticos y biólogos.
- Automatiza la instalación del cluster e incluye todo el software de HPC que un entusiasta del mismo necesita.
- Paquetes RPM y escritorio Gnome.
- Está disponible para las arquitecturas x86, IA64, y x86_64.
- Esta distribución es compatible con CentOS Linux y Red Hat Enterprise Linux.

Los softwares que utiliza son:

- MPICH
- LAM-MPI
- PVM
- Modules
- PVFS
- Myrinet GM
- Infiniband
- Sun Grid Engine
- Gcc
- Ganglio
- Globos

Aplicaciones actualizadas a 3.1:

- Actualiza los archivos del modulo a 1.0.2, incluyendo las correcciones a profile.d.
- Adiciona módulos con modulesenv 3.1.6
- Scripts sometidos por Humberto Zuazaga.
- Actualiza el hmmer a 2.3.2-1
- Actualiza el gromacs a 3.2.1-1
- Actualiza EMBOSS a 2.9.0-6; EMBOSS ahora instala a /usr/share y no /opt/BioBrew.
- Actualiza el NCBI BLAST con ncbiutils 6.1.0-2 y el ncbi.tar.gz
- Actualiza el mpiBLAST a 1.3; el mpiBLAST vive bajo / opt/NCBI/6.1.0/bin con el resto de los binarios.
- Actualiza Phylip a 3.61-5 y lo separa de EMBOSS. [4]

BioKnoppix

Características de BioKnoppix:

- Es una distribución personalizada del LiveCD de Knoppix Linux.
- Es una distribución completamente funcional del Sistema Operativo GNU/Linux con aplicaciones de código abierto orientada a la biología molecular.
- A pesar de que usa alguna memoria RAM, BioKnoppix no utiliza la computadora donde está corriendo, siendo de esta forma ideal para demostraciones, talleres, para ser utilizadas por estudiantes de biología molecular, etc.
- Un "live" con herramientas para biólogos moleculares.
- Paquetes DEB y escritorio KDE.

¿Cómo instalar BioKnoppix?

Inserte el CD y reinicie la computadora. Algunas computadoras requieren cambiar la opción del BIOS para despertar desde el CD. También existe una imagen de floppy sobre el CD que usted puede utilizar para despertar la computadora. Cuando usted cierra el Knoppix, el CD sale automáticamente.

BioKnoppix requiere:

- Una CPU compatible (o más de i486).
- 20 MB de RAM para modo texto, al menos 96 MB de modo gráfico con KDE (al menos 128 MB de RAM es recomendable el uso de varios productos office).
- Un Mouse estándar PS/2.
- Aplicaciones instaladas en BioKnoppix:
- EMBOSS: Suite europea de código abierto para Biología Molecular.
- Artemis: Browser libre para GENOME y herramienta de anotación.
- ImageJ: Herramienta de procesamiento de imágenes patrocinada por NIH. [5]

SaluX

Biolinux es un grupo de profesionales multidisciplinario, de difusión, investigación, desarrollo e implementación de Tecnologías Libres en el Área de la Salud que viene avanzando desde Agosto del año 2002.

El mismo se ha empeñado la tarea de desarrollar un sistema operativo para hospitales bajo los estándares y herramientas de free libre open-source software (FLOSS) para implementar en instituciones y centros de salud asistenciales (hospitales públicos y privados) de Argentina y, extensivamente, en Latinoamérica. Por lo tanto, nace en el año 2003 el Proyecto SaluX, constituyéndose en la distribución oficial del grupo Biolinux orientado al área de salud, especialmente diseñada para correr en centros hospitalarios, la cual esta basada en el kernel GNU/Linux y portadora de paquetes especialmente diseñados para instituciones hospitalarias.

Los principales objetivos de este Programa de Desarrollo son:

- Disponer de una distribución con elementos base informáticos para hospitales.
- Apostar a una paridad informática hospitalaria en Latinoamérica.
- Estimular las capacidades locales a través de un elemento informático de código abierto que pueda ser personalizable y escalable.
- Fomentar el uso del Software Libre en el área de salud en Latinoamérica.

- Mejorar las tecnologías de comunicación, incentivando el conocimiento y la coparticipación en efectores de salud rural. [6]

Las principales características de Salux son:

- Utiliza un kernel de la serie 2.6.x.x.
- Es un sistema operativo basado en Debian.
- Instalación grafica simplificada.
- Idioma: Español. (más adelante, portugués e inglés).
- Acceso sencillo a Internet y redes.
- Integración con otros sistemas existentes en red como las plataformas MS-Windows, Novell y Unix.
- Conectividad con redes cableadas, WiFi, Bluetooth, IrDA.
- Sistema de paquetes actualizables desde repositorios en Internet.
- Capacidad de terminal server, para trabajo con clientes livianos.
- Paquetes base: Navegador Mozilla Firefox, cliente de correo electrónico Mozilla Thunderbird, Gaim, Gimp, Kivio, Dia, Quanta, Bluefish, OpenOffice, etc., entre otras aplicaciones preinstaladas, así como el entorno de escritorio basado en KDE denominado SDE (SaluX Desktop Enviroment). Además SaluX podrá portar las aplicaciones destacadas en la Comunidad salud Libre como: OpenHIS, Castor, i-Path y aplicaciones destacadas en el portal Debian -Med.

En este proyecto se han desarrollado las siguientes versiones:

- SaluX Live CD
- SaluX MedStation (SX)
- SaluX MedServer (MS)
- Salux Terminal Server (TS)
- SaluX MTS

SaluX se constituye en un portador de sistemas informáticos de código abierto de avanzado desarrollo a nivel internacional en las áreas de telemedicina, medicina rural, sistemas de registro médico electrónico, etc. Además de paquetes de un sistema informático hospitalario en fase actual de desarrollo en el grupo Biolinux (OpenHIS) como parte integral del Proyecto Salud Libre. [7]

DNA Linux

El principal objetivo de esta distribución es que está orientada a la bioinformática, de Genes Digitales.

Las principales características de DNA Linux son:

- Está basada en una versión reducida de Slackware Linux (Slax).
- Incluye software de bioinformática y bases de datos genómicas.
- Funciona como un LiveCD, esto es, sin hacer cambios permanentes sobre el disco rígido.
- Puede ser instalado como un GNU/Linux regular.
- La distribución puede ser utilizada sin ningún riesgo de pérdida de datos.
- Actualmente está operativa la versión 0.5 Beta.
- El software de manipulación de la secuencia de DNA y de la proteína es incluido.
- Soporte para español, portugués y francés (inglés es el idioma que viene predeterminado).
- No está diseñada para utilizarse como un sistema operativo de “desktop”, sino como un servidor.

Avances de DNA Linux:

- Actualización del BLAST (2.2.10), EMBOSS (3.0.0), FinchTV, Clustal (1.83) y JRE (1.5).
- Programas gráficos a los que se puede acceder desde KDE.
- Basado en Slax 5.06 (con Kernel 2.6 y un mejor soporte del hardware).
- Instalador HD (no testado, usarlo con cuidado)
- Menor tamaño gracias a la mejor compresión provista por SquashFS [8]

Los softwares que utiliza son:

- BLAST
- EMBOSS-2.8.0
- T-COFFEE_distribution_Version_1.37
- abacus-0.63
- arka-0.11
- avid

- big_mac
- cap3
- clustalx1.81.linux
- e-pcr
- efmc
- hmmer-2.3.2.bin.intel-linux
- ncbi
- paml3.12
- primer3_0_9_test
- GeneSplicer

Linux Hélice

Esta distribución de GNU/Linux es un proyecto colombiano, creado por Andrés M. Pinzón. Linux Hélice (LH) es una distribución GNU/Linux (en formato de LiveCD), inicialmente pensada como una solución a las necesidades informáticas, presupuesto e infraestructura de la mayoría de laboratorios de biología molecular en Colombia.

Sin embargo, actualmente su desarrollo está enfocado completamente hacia la educación y divulgación, tanto de la Bioinformática como disciplina anexa a la biología, como del software.

Objetivo de Linux Hélice

- Centralizar el análisis de datos biológicos localmente, evitando al máximo todo tipo de análisis por la red.

Características de Linux Hélice

- Se empezó a desarrollar con "Linux from Scratch" pero varios intentos se vieron frustrados, tanto por la complejidad del proceso como por la inutilidad del mismo.
- Después su base fue Scilix, pero en vista de que el proyecto Tangram Linux empezaba a tomar vuelo, se decidió optar por que LH fuera un "mini-módulo de biología" dentro de dicho proyecto.

- Está pensada especialmente en la infraestructura de redes y hardware típicos de los laboratorios en los países menos desarrollados.
- Además de ofrecer servicios de soporte y manuales del software presente en la distribución, Linux Hélice centra sus esfuerzos en la implementación de interfaces gráficas para la mayoría de programas que funcionan en consola.

Avances de Linux Hélice:

- En estos momentos está desarrollando una versión "re-contrá" beta de LH, basada en Slackware, la cual ha dado buenos resultados, pero que a mediano plazo podría resultar ineficiente. [9]

En la actualidad:

- Esta distribución se encuentra un poco desactualizada.

Estas distribuciones resultan de gran ayuda; sin embargo no resuelven todas las necesidades existentes hasta el momento. El problema fundamental que presentan radica en que están muy enfocadas al trabajo con Biología Molecular y dejan a un lado las restantes áreas de la Bioinformática, como por ejemplo: la Química Teórica, la Biología de Sistemas y el Diseño de Fármacos. Además, para el trabajo con muchas de estas distribuciones se requiere de un conocimiento mínimo del uso del sistema operativo GNU/Linux, lo que ha devenido en una traba, pues muchos miembros de la comunidad científica temen que la migración a GNU/Linux atrase sus investigaciones, por tener que enfrentarse con un Sistema Operativo difícil de manejar.

Como solución a estos problemas se ha planteado la necesidad de crear una nueva distribución que se caracterice por su amigabilidad con el usuario, que incluya paquetes de aplicaciones para las diferentes ramas de la Bioinformática y que a su vez ponga en manos de los científicos un grupo de herramientas para muchos desconocidas, que le pueden permitir acelerar las investigaciones en esta área.

Se pretende crear una nueva distribución en forma de LiveCD específica para la Bioinformática.

A continuación se detallan las principales metadistribuciones existentes con el objetivo de seleccionar aquella que resulte más conveniente para el desarrollo de la distribución deseada.

Principales distribuciones de GNU/Linux

Varias son las distribuciones de GNU/Linux que se han creado a nivel mundial gracias al trabajo constante de los desarrolladores y promotores del Software Libre en el mundo entero. Ejemplo de ellas son: Debian, Mandriva, Ubuntu, Novel/Suse, Red Hat y Gentoo. La gran variedad existente motiva grandes contradicciones entre los usuarios, quienes no saben por cual decidirse. Analizar las características de cada una de estas distribuciones puede resultar de gran ayuda para aliviar estas contradicciones, si se tiene en cuenta que de ella se desprende la utilidad que pueda tener en determinada rama.

Debian

Debian o, más exactamente, Debian GNU/Linux es una distribución GNU/Linux que basa sus principios y fin en el Software Libre. Fue creada por el proyecto Debian en el año 1993, la organización responsable de la creación y mantenimiento de la misma distribución, centrado en GNU/Linux y utilidades GNU.

Nace como una apuesta por separar en sus versiones el Software Libre del Software no Libre. El modelo de desarrollo es independiente a empresas, creado por los propios usuarios, sin depender de ninguna manera de necesidades comerciales. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

La primera adaptación del sistema Debian, siendo también la más desarrollada es Debian GNU/Linux, basada en el núcleo Linux. Existen también otras adaptaciones con núcleos Hurd (Debian GNU/Hurd), NetBSD, (Debian GNU/NetBSD) y FreeBSD (Debian GNU/FreeBSD).

Distribuciones basadas en Debian:

En la actualidad, como Debian es una distro que ha probado su estabilidad y utilidad, muchos desarrolladores la han tomado para crear nuevas distribuciones, como es el caso de:

- Knoppix: distribución LiveCD con muchos derivados (tales como Morphix, Damn Small Linux, Knotix y KnoppMyth).
- Linspire y Freespire: distribuciones diseñadas específicamente para usuarios sin experiencia técnica.
- Mepis: distribución LiveCD para novatos, que se puede también instalar.
- Ubuntu: disponible también como LiveCd y en muchas versiones, entre ellas Kubuntu, Xubuntu, Edubuntu, y qNewSence (apoyada por la FSF, Free Software Foundation/Fundación del Software Libre).
- Xandros un derivado de Corel Linux con asistencia comercial.
- RXART: Desarrollada en Argentina especial para iniciarse, distribución comercial.
- Nexenta OS: un sistema operativo utilizando el kernel de Opensolaris con las librerías GNU. (Alpha).
- Hasta Enero de 2007 se reportó que hay 129 distribuciones basadas en Debian. [10]

Características de Debian:

- La disponibilidad en varias plataformas hardware. La versión 3.1a es compatible con 11 plataformas.
- Una amplia colección de software disponible. La versión 3.1a viene con unos 15490 paquetes.
- Un grupo de herramientas para facilitar el proceso de instalación y actualización del software.
- Su compromiso con los principios y valores involucrados en el movimiento de Software Libre.
- No tiene marcado ningún entorno gráfico en especial, ya sea GNOME, KDE u otro.

Desventajas de Debian:

Debian resulta ser difícil de instalar, a menos que el usuario tenga un profundo conocimiento del hardware de su computadora.

Últimas noticias

La última versión estable de Debian es la 3.1. La última actualización de esta versión se publicó el 18 de febrero de 2007.

- [18 de feb de 2007] Debian GNU/Linux 3.1 update
- [10 de ene de 2007] Debian GNU/Linux 3.0 archivo
- [6 de nov de 2006] Debian GNU/Linux 3.1 actualizado
- [1 de sep de 2006] Debian GNU/Linux 3.1 actualizado
- [3 de ago de 2006] Extremadura adopta Debian GNU/Linux
- [24 de jul de 2006] La siguiente versión de Debian GNU/Linux, 4.0. [11]

Mandriva

Mandriva es una distribución de GNU/Linux surgida como resultado de la fusión entre dos compañías de GNU/Linux: Mandrake de origen francés y Conectivo de origen brasileña. Mandriva Linux (antes Mandrakelinux y Mandrake Linux) es una distribución GNU/Linux aparecida en julio de 1998 propiedad de Mandriva, enfocada a principiantes o usuarios medios. Todo el código de Mandriva es publicado bajo la Licencia Pública General (GPL).

La primera edición se basó en Red Hat Linux (versión 5.1) y escogió el entorno gráfico de KDE (versión 1.0). Desde entonces ha seguido su propio camino, separado de RedHat y ha incluido numerosas herramientas propias o modificadas, fundamentalmente dirigidas a facilitar la configuración del sistema. Mandrake (su anterior nombre) también es conocida por compilar sus paquetes con optimizaciones para procesadores Pentium y superiores, incompatibles con versiones más antiguas tales como 386 y 486.

Características de Mandriva:

- Mandriva Linux está disponible en unos 74 idiomas. Especialmente de calidad son sus traducciones al español, catalán y portugués.
- El instalador de Mandriva Linux es, probablemente, el más amigable de entre las diferentes distribuciones de GNU/Linux.
- El instalador está traducido a más de 70 idiomas.
- Mandriva Linux emplea mandrake Control Center para la administración de GNU/Linux, en lugar de un editor de texto para cambiar aspectos de la configuración.

- Mandriva Linux, que forma parte del grupo LSB (Linux Standard Base), viene con 12.306 paquetes de software (versión 2006), incluyendo juegos, programas de oficina, servidores y utilidades de Internet.
- No se basa en un único entorno de escritorio. Proporcionando apoyo tanto a KDE (QT) como a Gnome (GTK), apoyando tanto el desarrollo de programas QT (Kat, buscador integrado en KDE) como GTK (las herramientas de administración de Mandriva están escritas en GTK).
- Utiliza Urpmi para la administración de programas.
- Su avanzado entorno gráfico y su centro de control nos permite hacer prácticamente todo lo que se necesite sin necesidad de abrir un terminal y "escribir los comandos" que tanto nos asustan al principio.
- Mandriva 2007, incorpora un escritorio 3D, ya sea con un programa llamado Compiz o con Beril. Lo que posibilita tener 4 escritorios formando un cubo y poder girarlo sobre sí mismo para cambiar de escritorio.

¿Quiénes utilizan Mandriva Linux?

- Usuarios individuales sin conocimiento específico de TI que necesitan de una distribución que sea sencilla de utilizar.
- Usuarios experimentados que prefieren no gastar las horas instalando y configurando su equipo ya que Mandriva Linux no requiere de esfuerzo para instalar y mantener. Esta distribución permite su gestión tanto con asistentes, como con editores de texto a la vieja usanza.
- Profesionales que necesitan sistemas poderosos, amplios y estables. Mandriva Linux es una de las más completas distribuciones GNU/Linux y está volviéndose una de las más favoritas en negocios. [12]

Ubuntu

El 8 de julio de 2004 fue anunciada la creación de la distribución Ubuntu. Distribución de GNU/Linux que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio. Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad en la restricción de uso, los lanzamientos regulares y la facilidad en la instalación. Ubuntu es patrocinado por Canonical Ltd, una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

La versión más reciente, Ubuntu 6.10 (Edgy Eft), fue lanzada el 26 de octubre de 2006. La versión 7.04 (Feisty Fawn) está programada para ser lanzada el 19 de abril de 2007.

Características de Ubuntu:

- Basada en la distribución Debian.
- Disponible en 4 arquitecturas: Intel x86, AMD64, PowerPC, SPARC (para esta última sólo existe la versión servidor).
- Los desarrolladores de Ubuntu se basan en gran medida en el trabajo de las comunidades de Debian y GNOME.
- Las versiones estables se liberan cada 6 meses y se mantienen actualizadas en materia de seguridad hasta 18 meses después de su lanzamiento.
- La nomenclatura de las versiones no obedece principalmente a un orden de desarrollo, se compone del dígito del año de emisión y del mes en que esto ocurre. La versión 4.10 es de octubre de 2004, la 5.04 es de abril de 2005, la 5.10 de octubre de 2005, la 6.06 es de junio de 2006 y la 6.10 es de octubre de 2006.
- El entorno de escritorio oficial es Gnome y se sincronizan con sus liberaciones.
- Para centrarse en solucionar raudamente los bugs, conflictos de paquetes, etc. se decidió eliminar ciertos paquetes del componente main, ya que no son populares o simplemente se escogieron de forma arbitraria por gusto o sus bases de apoyo al Software Libre. Por tales motivos inicialmente KDE no se encontraba con más soporte de lo que entregaban los mantenedores de Debian en sus repositorios, razón por la que se sumó la comunidad de KDE distribuyendo una distribución llamada Kubuntu.
- De forma sincronizada a la versión 6.06 de Ubuntu, apareció por primera vez la distribución Xubuntu, basada en el entorno de escritorio XFce.
- El navegador web oficial es Mozilla Fire fox.
- El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse. Por eso mismo, no hay un firewall predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema.
- Para labores/tareas administrativas en terminal incluye una herramienta llamada sudo (similar al Mac OS X), con la que se evita el uso del usuario root (administrador).

- Mejora la accesibilidad y la internacionalización, de modo que el software esté disponible para tanta gente como sea posible. En la versión 5.04, el UTF-8 es la codificación de caracteres en forma predeterminada.
- No sólo tiene como lazo a Debian el uso del mismo formato de paquetes deb, Ubuntu tiene uniones muy fuertes con esa comunidad, contribuyendo cualquier cambio directamente e inmediatamente, más que anunciándolos. Esto sucede en los tiempos de lanzamiento. Muchos de los desarrolladores de Ubuntu son también responsables de los paquetes importantes dentro de la distribución de Debian.
- Todos los lanzamientos de Ubuntu se proporcionan sin costo alguno. Los CDs de la distribución se envían de forma gratuita a cualquier persona que los solicite mediante el servicio Shiplt (la versión 6.10 ya no se distribuye en cd's, la versión anterior se sigue distribuyendo sin costo alguno). También es posible descargar las imágenes ISO de los discos por transferencia directa o bajo la tecnología Bittorrent.
- Ubuntu no cobra honorarios por la suscripción de las mejoras de la "Edición Enterprise".

Variantes:

Existen diversas variantes de Ubuntu disponibles, las cuales poseen lanzamientos simultáneos con Ubuntu. Las más significativas son:

- Kubuntu, el cuál utiliza KDE en vez de GNOME.
- Edubuntu, diseñado para entornos escolares.
- Xubuntu, el cuál utiliza el entorno de escritorio Xfce.

Kubuntu, Edubuntu y Xubuntu son proyectos oficiales de la Ubuntu Foundation. Kubuntu y Edubuntu se encuentran incluidos dentro del programa Shiplt.

Otras distribuciones GNU/Linux basadas en Ubuntu son:

- Fluxbuntu – Fluxbox.
- Ubuntu Christian Edition - Para usuarios y comunidades cristianas.
- Ebuntu – Enlightenment.

EUbuntu - Dedicado a temas de seguridad y redes (lleva Fluxbox como gestor de ventanas).

Ubuntu Lite - Optimizado para máquinas lentas.

Ubuntu Studio.

Dubuntu - Para desarrolladores

Admelix

Agnula

BARDINUX - Distribución Universidad de La Laguna (Tenerife)

BIG LINUX

EHUX

GNIX

Guadalinex

Impi

Linux Mint

Mepis (Desde abril de 2006)

Molinux.

Ubuntu Christian edition - Con software cristiano

Ubuntu Muslim Edition - Con software islámico

Ufficio

Urli [13]

Suse

SUSE Linux es una de las más conocidas distribuciones GNU/Linux existentes a nivel mundial.

Características de Suse:

- Suse está basada en Slackware.
- Suse es sencilla de instalar y administrar gracias a la presencia de los asistentes gráficos.
- Incluye un programa único de instalación y administración llamado YaST2 que permite realizar actualizaciones, configurar la red y el cortafuego, administrar a los usuarios, y muchas más opciones todas ellas integradas en una sola interfaz.

- Incluye varios escritorios, entre ellos los más conocidos que son KDE y Gnome, siendo el primero el escritorio por omisión.
- La distribución incorpora las herramientas necesarias para redistribuir el espacio de tu disco duro permitiendo así la coexistencia con otros sistemas operativos existentes en el mismo.
- Usa sistemas de paquetes RPM.

Versiones publicadas:

1.0 - Marzo 1994, 2.0 - ????, 3.0 - 1995, 4.0 - 1996, 5.0 - Noviembre 1997, 6.0 - Enero 1999, 7.0 - Septiembre 2000, 8.0 - Abril 2002, 9.0 - Octubre 2003, 10.0 - Septiembre 2005, 10.1 - Mayo 2006, 10.2 - Diciembre 2006.

Distribuciones GNU/Linux basadas en SUSE:

Linux Novell, Linux Desktop, Java Desktop, System SUSE Linux Enterprise Edition. [14]

Red Hat

Red Hat Linux es una distribución GNU/Linux creada por Red Hat, la cual fue una de las más populares en los entornos de usuarios domésticos.

La versión 1.0 de esta distro fue presentada el 3 de noviembre de 1994 por Bob Young y Marck Swing. No es tan antigua como la distribución Slackware, pero ciertamente es más antigua que muchas otras. Fue la primera distribución que usó RPM como su formato de paquete, y en un cierto plazo ha servido como el punto de partida para varias otras distribuciones, tales como la orientada hacia PCs de escritorio Mandrake Linux (originalmente Red Hat Linux con KDE), Yellow Dog Linux, la cual se inició desde Red Hat Linux con soporte para PowerPC, y ASPLinux (Red Hat Linux con mejor soporte para caracteres no-Latinos).

Desde el 2003, Red Hat ha desplazado su enfoque hacia el mercado de los negocios con la distribución Red Hat Enterprise Linux y la versión no comercial Fedora Core.

Red Hat Enterprise Linux también conocido por sus siglas RHEL es una distribución comercial de GNU/Linux desarrollada por Red Hat. Es la versión comercial de Fedora Core, y anteriormente lo era de RH Linux, de forma similar a como Novell SUSE Enterprise lo es respecto de OpenSUSE.

Mientras que las nuevas versiones de Fedora salen cada aproximadamente 6 meses, las de RHEL suelen hacerlo cada 18 o 24 meses. Tienen soporte oficial de Red Hat, programas de certificación, etc.

Fedora Core (también conocida como Fedora Linux) es una distribución GNU/Linux desarrollada por la comunidad Fedora y promovida por la compañía estadounidense Red Hat.

El objetivo del proyecto Fedora es conseguir un sistema operativo de propósito general y basado exclusivamente en Software Libre con el apoyo de la comunidad GNU/Linux. Los ingenieros de Red Hat continúan participando en la construcción y desarrollo de este proyecto e invitan y fomentan la participación de miembros de la comunidad Linux.

Características de Red Hat

Programadores empleados de Red Hat han desarrollado múltiples paquetes de Software Libre, los cuales han beneficiado a toda la comunidad.

- Algunas de las contribuciones más notables han sido la creación de un sistema de empaquetación de software (RPM), y varias utilidades para la administración y configuración de equipos, como `snodconfig` o `mouseconfig`.
- Algunas de las distribuciones basadas en RedHat Linux más importantes son: Mandrake Linux y Yellow Dog Linux (sólo para PowerPC).

Red Hat Enterprise Linux, la primera solución para la computación de código abierto, se ofrece en productos que van desde el escritorio hasta el centro de datos:

- Red Hat Enterprise Linux AS — para sistemas high-end y de misión crítica
- Red Hat Enterprise Linux ES — para servidores de pequeño y mediano rango
- Red Hat Enterprise Linux WS — para estaciones de trabajo técnicas/diseño en sistemas cliente

- Red Hat Desktop — para clientes seguros y manejables [15]

Gentoo

En el año 1999 Daniel Robbins se propuso crear una nueva distribución de GNU/Linux, a pesar de no contar con suficientes recursos que le permitieran competir con Debian o Red Hat. Por tal motivo se decidió a automatizar el proceso de crear los paquetes lo más comprimidos posible. La distribución se crearía a sí misma desde cero; los usuarios seguirían el mismo proceso y compilarían todos sus binarios, eliminando la grasa superflua y quedándose con un sistema optimizado al máximo. El motivo de que esta distribución se tardara tanto en aparecer, se debió a sus propias particularidades.

El nombre de Gentoo viene de uno de los pingüinos más veloces en el agua, sin duda refiriéndose a lo rápido que van los binarios compilados para nuestra máquina y nuestras necesidades. Esta distribución no se limita a empaquetar unos cuantos programas específicos de algún campo. Más bien intenta ser una distribución universal; y con más de 10,000 paquetes y dando soporte a 6 arquitecturas. [16]

Ventajas de Gentoo

- Tiene un sistema totalmente optimizado y personalizado. Mediante el fichero make.conf se definen los criterios por defecto a la hora de compilar, como librerías o arquitectura del ordenador a usar. Así, si se tiene un ordenador antiguo sin grabadora ni nada, los programas se instalaran sin soporte para la misma ahorrando tanto espacio como recursos.
- El kernel, el cual también se tiene que configurar, tiene soporte solo para lo que le interese al usuario. Además se pueden poner los módulos enviados, ahorrando tiempo de carga.
- Gentoo, además, hay que instalarlo a piezas desde cero, por lo que no se tendrán programas innecesarios, lo que significa que se puede decidir cual usar. No tiene nada “por defecto”, pudiendo elegir cualquier gestor de ventanas si necesidad de bajar versiones específicas.
- A la hora de compilar, Gentoo le da la posibilidad al usuario de poder hacer que los programas se adapten a su gusto y personalizar cada paquete de forma individual.
- Gentoo Cuenta con grandes repositorios, los cuales son actualizados con gran rapidez. Al usar códigos fuente resulta muy fácil añadir nuevos programas. De aun no tener suficiente, podremos

recurrir al overlay, que son los no oficiales. Debido al sistema tan sencillo de gestión de paquetes mediante ebuilds (pequeños documentos con las dependencias del programa, lugar de descarga y pequeño scrip de instalación), puedes bajarte un ebuild casero y añadirlo al repositorio sin muchos problemas. [17]

Desventajas de Gentoo

- La principal ventaja de Gentoo también es su principal desventaja. El hecho de tener que compilar todo puede volverse muy cansado, pues generalmente es un proceso lento que puede llegar a tardar hasta varias horas por paquetes (openoffice por ejemplo).
- Hay que instalar a mano prácticamente todo, y algo tan sencillo como configurar el ratón puede volverse una locura si no se ha configurado correctamente el kernel.
- Es un sistema totalmente vacío que se tendrá que rellenar, y cualquier cosa, por simple y tonta que parezca, si no se configura, no estará. Da igual que parezca algo fundamental, habrá que instalarlo, como el cliente dhcp o los drivers del teclado para las X. Esto puede provocar grandes errores que le pueden hacer perder al usuario gran cantidad de tiempo. [17]
- Poner en marcha un sistema completo, o actualizar un sistema que ha estado desatendido durante una temporada, puede requerir una respetable cantidad de tiempo (horas o incluso días si el ordenador es muy antiguo), mientras se descargan y compilan todos los paquetes nuevos.

A pesar de las desventajas se puede concluir que Gentoo permite por regla general una actualización sin problemas, a diferencia de otras distribuciones donde puede llegar a resultar complicado o casi imposible. Esta actualización también es posible a partir de binarios precompilados, lo que requiere menos tiempo.

Gentoo da una sensación de dinamismo controlado bastante curiosa: está siempre a medio hacer, y aún así consigue generar sistemas estables con las últimas versiones de los paquetes. Es un hecho que cada día son más los usuarios que se deciden a instalar esta distribución en su computadora y parece que muchos de ellos terminan dejándola ahí, quitando el protagonismo a su anterior distribución. Su aproximación minimalista a la configuración de paquetes permite saber exactamente qué realmente se tiene en la máquina.

Si se quiere un sistema adaptado al gusto particular de cada cual y se tienen algunos conocimientos previos de GNU/Linux, Gentoo es la distribución ideal para lograrlo.

Características específicas:

- Posee un sistema de fabricación de binarios, ebuild, el cual tiene su origen en el software que diseñó Robbins para ayudarle a hacer su primera distribución; además de los flags para el compilador, se puede elegir qué dependencias incluir, globalmente y para paquetes individuales. Esto evita dependencias a interfaces determinadas y software no deseados por el usuario (como por ejemplo el software "arts").
- Otra novedad que posee a nivel mundial es Portage. Se trata de una grandísima colección de software indexado, categorizado y aún mejor: verificado. El sistema se quejará y no dejará instalar los paquetes que estén en estado masked, es decir "enmascarados". Ciertos programas no funcionan en algunas arquitecturas (por ejemplo, OpenOffice.org versión 1.x no compilaba para amd64). Otros no están todavía probados; se quedan en cuarentena hasta que se verifique que funcionan. Hay incluso paquetes incompatibles entre sí: no se pueden tener varios servidores de correo andando al mismo tiempo. Así, cuando el software pedido no cumple ciertas condiciones, se "enmascara" para que no cause problemas.
- Posee una herramienta que hace que el trabajo de compilar el núcleo del sistema sea mínima, Genkernel, la cual hace poco fue portada para otras distribuciones como Debian, y que tiene la característica de que autodetecta el hardware y automáticamente construye un kernel específico para este, además, en caso de que se quieran agregar módulos adicionales que provean determinado soporte, se puede hacer mediante un menú en modo gráfico.

¿Quiénes utilizan Gentoo?

Gentoo Linux es una distribución GNU/Linux orientada a usuarios con cierta experiencia en este sistema operativo.

Distribuciones basadas en Gentoo:

Bintoo
Flash Linux
Gentoox
Knopperdisk
Kororaa
Librix
Medeix
Navyn OS
Pardus
Pentoo
Mayix
RR4/RR64
SabayonLinux
SystemRescue
Ututo
Vidalinux [16]

Distribución a utilizar como punto de partida

Para la realización de este trabajo se utilizó como distribución de partida a Nova. Esta distribución, realizada por un grupo de estudiantes de la Universidad de las Ciencias Informáticas, está basada en Gentoo y surge a mediados del 2005 como respuesta a la necesidad de una plataforma que garantiza la compatibilidad de las aplicaciones que están en desarrollo con sistemas libres.

La primera característica que llevó a escoger esta distribución, fue la base que utilizó para su desarrollo. Nova, al estar basada en Gentoo aprovecha todas sus facilidades y flexibilidad, además de que a pesar de ser una metadistribución muy joven posee características muy interesantes y fáciles de aprovechar. Gentoo es una distribución muy adaptable a los diferentes tipos de hardware, lo que posibilita su buen funcionamiento sobre diferentes arquitecturas; además sus repositorios se mantienen muy actualizados y la mayoría de los paquetes seleccionados ya están en los mismos. Por otro lado, Gentoo es una distribución generada a partir de código fuente lo que le permite brindar la posibilidad de ser flexible en

cuanto a su personalización; hecho que aprovecha Nova y que por transitividad va a utilizar Nova BioLive, en este caso con aplicaciones puramente bioinformáticas.

Otra de las características que hizo que se escogiera esta distribución como base fue la de aprovechar el trabajo realizado por el equipo de desarrollo de la facultad 10, poniendo de manifiesto los principios éticos del Software Libre que es la solidaridad, la ayuda mutua, el trabajo en conjunto, en este caso, entre dos facultades, donde una de estas se especializa en el desarrollo del Software Libre y los propósitos de la otra van encaminados a satisfacer y darle solución a alguno de los problemas que existan en el ámbito de la Bioinformática.

Nova fue mucho más allá de la personalización de una distribución de GNU/Linux, se logró una plataforma de desarrollo y trabajo adaptable a todos los entornos posibles y mantiene una línea orientada a escritorio y otra a servidores, con facilidades para su instalación, mejoras para incrementar su rendimiento, herramientas para lograr una transición a GNU/Linux menos traumática y por sobre todo la capacidad de convertirse en lo que un cliente específicamente necesita.

Nova además construyó un repositorio de paquetes binarios para reducir el tiempo del proceso de instalación y se desarrolló un instalador de paquetes usando la combinación Gtk # y Mono, el cual es sustituido más tarde por una versión modificada del Porthole, bautizada NEIN (No Es el Instalador de Nova).

En estos momentos Nova cuenta con una herramienta que permite manejar recursos compartidos usando el protocolo Samba, así como cambiar la contraseña del dominio y una herramienta para el manejo de particiones, las cuales se integrarán en el Centro de Control, actualmente en desarrollo, el cual permitirá las funcionalidades más comunes de administración y configuración.

Hasta ahora el desarrollo de Nova se ha centrado principalmente en las características internas de la Universidad de Ciencias Informáticas, obteniéndose varias imágenes para estaciones de trabajo, como son: imagen para docencia, imagen para entornos de oficina y imagen para la Copa Void de Programación.

Estas imágenes han sido construidas exclusivamente para el hardware existente en la universidad y el software incluido se ha compilado con opciones cuidadosamente escogidas, con el fin de optimizar el funcionamiento se ha suprimido el soporte innecesario de los binarios obtenidos. Se ha utilizado el entorno gráfico Gnome, eliminando de todo el sistema el soporte a KDE o QT, de esta forma los ejecutables obtenidos son más pequeños y, por tanto, más rápidos a la hora de ser ejecutados. Para aumentar aun más el rendimiento general, se ha especificado en el kernel el nivel de prioridad con que se ejecutan los diferentes procesos aumentando la importancia de los vinculados a programas que interactúan directamente con el usuario.

Se puede descargar el ISO de un LiveCD optimizado para lograr mejoras de rendimiento y un mínimo de aplicaciones, ideal para trabajos de oficina simples, usando XFCE como entorno gráfico.

Por ultimo se puede decir, que debido a esta selección podemos aprovechar el trabajo realizado por el equipo de desarrollo de la distribución Nova, lo que facilitaría el trabajo pues ya no habría que empezar desde cero, sino que se remasterizaría Nova.

Capítulo 2: Descripción del Proceso de Desarrollo de una Distribución de GNU/Linux

Introducción

Este capítulo está dedicado a la descripción del proceso de desarrollo de una distribución de GNU/Linux en general. Su objetivo fundamental es ofrecer una metodología para la creación de una distribución de GNU/Linux, aunque ofrece un estudio bastante amplio del estado actual de la Ingeniería del Software Libre y ofrece datos relevantes acerca de la creación de LiveCD.

Metodología general para la creación de una distribución de GNU/Linux.

Para el desarrollo de una distribución de GNU/Linux es necesario seguir una serie de pasos partiendo de los objetivos que se persigue con la realización de la misma. Es importante tener bien definido estos objetivos para encaminar el esfuerzo al resultado final en dependencia de las especificidades del usuario para el que se está trabajando.

Acorde a los objetivos previstos, se seleccionan las aplicaciones a incluir en la distribución final y el instalador de aplicaciones que va a depender de la distribución base sobre la cual se vaya a trabajar; ejemplo: rpm en red hat, apt en debian, tbz2 en gentoo, pkg en slackware, etc.

Posteriormente se procede a la búsqueda o creación de las herramientas para automatizar los procesos. Existen varias herramientas para esto, casi siempre cada distribución crea la suya propia, en dependencia de los recursos que tengan a mano y de los lenguajes de programación que sepan utilizar los que se encargaran de crearla. El kit, que es como se le llama a dichas herramientas, permite automatizar el proceso de personalizar el código fuente, compilarlo, generar el paquete que permitirá instalarlo y ponerlo en el repositorio, luego un módulo de éste puede ser utilizado para crear el CD de instalación a partir de la lista de aplicaciones que necesita el instalador, el propio instalador y una imagen básica del sistema.

Estas herramientas se encargaran de crear el repositorio y el CD de instalación. Para esto se seleccionara solo lo necesario siempre y cuando funcione y cumpla con todas las expectativas del usuario.

Conjuntamente con esto se pueden crear otros productos como un LiveCD, LiveCDUSB (la única diferencia con los LiveCD es que pueden guardar datos) y otros productos a la medida como routers, switch y celulares.

Por último y no por ello menos importante, se procede a la creación de la infraestructura de soporte. Esta infraestructura permite mantener un desarrollo estable, sostenido y progresivo de la distribución, permitiendo brindar información (portal o cms); mantener una documentación de las aplicaciones, el sistema y su utilización y una comunicación fluida y estable con los usuarios del sistema (wiki y foro); reporte y gestión de incidencias (bugzilla); los repositorios de paquetes, ISOS y otros archivos; espejos de los repositorios.

Ingeniería del Software Libre.

Aunque hace ya varias décadas que se desarrolla Software Libre, sólo desde hace unos pocos años se ha empezado a prestar atención a sus modelos y procesos de desarrollo desde el punto de vista de la Ingeniería del Software. Igual que no hay un único modelo de desarrollo de Software Propietario, tampoco lo hay en el mundo del Software Libre. Aunque pueden encontrarse interesantes características que comparten gran parte de los proyectos estudiados, y que podrían estar enraizadas en las propiedades de los programas Libres.

En 1997, Eric S. Raymond publicó el primer artículo ampliamente difundido, “La catedral y el bazar”, donde se trataba de describir algunas características de los modelos de desarrollo de Software Libre, haciendo especial énfasis en lo que diferencia estos modelos de los de desarrollo propietario.

En el Software Libre, a diferencia del Software Propietario, no existen roles ni tareas establecidas con anterioridad, sino que los roles tienen la propiedad de cambiar de manera continua sin indicación externa.

Lo más novedoso de “La catedral y el bazar” es la descripción del proceso que ha hecho de GNU/Linux un éxito dentro del mundo del Software Libre; es una sucesión de buenas maneras para aprovechar al máximo las posibilidades que ofrece la disponibilidad de código fuente y la interactividad mediante el uso de sistemas y herramientas telemáticas.

Un proyecto de Software Libre suele surgir a raíz de una acción puramente personal. Una vez que el desarrollador haya conseguido tener algo usable, con algo de funcionalidad, sencillo y, a ser posible, bien diseñado o escrito, lo mejor que puede hacer es compartir esa solución con la comunidad del Software Libre. A esto se le conoce como publicación temprana y que permite llamar la atención de otras personas (generalmente desarrolladores) que tengan el mismo problema y que puedan estar interesados en la solución.

Uno de los principios básicos de este modelo de desarrollo es considerar a los usuarios como codesarrolladores pues van a estar encargados de realizar las pruebas, una de las tareas más costosas en la generación de software. Al contrario que el codesarrollo, que es difícilmente escalable, la depuración y las pruebas tienen la propiedad de ser altamente paralelizables. El usuario será el que tome el software y lo pruebe en su máquina bajo unas condiciones específicas (una arquitectura, unas herramientas, etc.), una tarea que multiplicada por un gran número de arquitecturas y entornos supondría un gran esfuerzo para el equipo de desarrollo. Además si se trata a los usuarios como desarrolladores, puede darse el caso de que alguno de ellos encuentre un error y lo solucione o al menos informe de el para que el problema esté solucionado en la siguiente versión. Cualquiera de las dos opciones son de gran utilidad para el desarrollo del software.

Generalmente, el modelo de desarrollo en proyectos de Software Libre suele ser de una manera informal, debido a que gran parte del equipo de desarrollo realiza esas tareas de manera voluntaria y sin recompensa económica.

Fases de la Ingeniería de Software Libre:

- Captura de requisitos

La forma como se capturan requisitos en el mundo del Software Libre depende tanto de la edad como del tamaño del proyecto. En las primeras etapas, fundador de proyecto y usuario suelen coincidir en una misma persona. Más adelante, y si el proyecto crece, la captura de requisitos suele tener lugar a través de las listas de correo electrónico y se suele llegar a una clara distinción entre el equipo de desarrollo. Para proyectos grandes, con muchos usuarios y muchos desarrolladores, la captura de requisitos se hace mediante la misma herramienta que se utiliza para gestionar los errores del proyecto. En este caso, en vez

de hablar de errores, se refieren a actividades, aunque el mecanismo utilizado para su gestión es idéntico al de la corrección de errores. En cualquier caso, no suele ser común un documento que reúna los requisitos tal y como es normal en el modelo en cascada.

- **Diseño global del sistema**

Sólo los grandes proyectos suelen tener documentado el diseño de manera exhaustiva. En el resto de proyectos, lo más probable es que el o los desarrolladores principales sean los únicos que lo posean (a veces sólo en su mente), o que vaya fraguándose en versiones posteriores del software. La carencia de un diseño detallado no sólo implica limitaciones en cuanto a la posible reutilización de módulos, sino que también es un gran impedimento a la hora de permitir el acceso a nuevos desarrolladores, ya que éstos se tendrán que enfrentar a un proceso de aprendizaje lento y costoso. El diseño detallado, por su parte, tampoco está muy generalizado. Su ausencia implica que se pierdan muchas posibilidades de reutilización de código.

- **Implementación**

La implementación es la fase en la que se concentra el mayor esfuerzo por parte de los desarrolladores de Software Libre, entre otras razones porque a ojos de los desarrolladores es manifiestamente la más divertida. Para ello, se suele seguir el paradigma de programación clásico de prueba y error hasta que se consiguen los resultados apetecidos desde el punto de vista subjetivo del programador.

- **Prueba**

Las pruebas automáticas no están muy arraigadas. Por lo general serán los propios usuarios, dentro de la gran variedad de usos, arquitecturas y combinaciones, los que las realizarán. Esto presenta la ventaja de que se paralelizan a un coste mínimo para el equipo de desarrollo. El problema que plantea este modelo es cómo organizarse para que la realimentación por parte de los usuarios exista y resulte lo más eficiente posible.

- **Mantenimiento del software**

En cuanto al mantenimiento del software en el mundo del Software Libre –refiriéndonos con ello al mantenimiento de versiones antiguas, es una tarea cuya existencia depende del proyecto. En proyectos donde se requiere una gran estabilidad, como núcleos del sistema operativo, etc., se mantienen versiones antiguas de los mismos, ya que un cambio a una nueva versión puede resultar traumático. Pero, por lo

general, en la mayoría de los proyectos de Software Libre, si se tiene una versión antigua y se encuentra un error, los desarrolladores no suelen ponerse a corregirlo y aconsejan utilizar la versión más moderna con la esperanza de que el error haya desaparecido, ya que el software ha evolucionado.

- COCOMO

Es un modelo de estimación de esfuerzos clásico que data de principios de la década de los ochenta y que se utiliza fundamentalmente en proyectos de software propietario para realizar una estimación de esfuerzo, duración del proyecto y costes.

El modelo COCOMO toma como medida de entrada el número de líneas de código fuente y genera estimaciones de coste total, tiempo de desarrollo y esfuerzo dedicado. COCOMO es un modelo pensado para procesos de generación de software clásicos (desarrollo en cascada o en V) y para proyectos de tamaño medio o grande.

A pesar de ser un modelo para ser utilizado fundamentalmente por software propietario, el resultado que se obtenga tras su uso, puede dar una idea del orden de magnitud en el que nos movemos, con información sobre los esfuerzos óptimos necesarios si se hubiera utilizado un modelo de desarrollo propietario.

En general, lo que más asombra de los resultados de COCOMO es su estimación de costes. En dicha estimación se tienen en cuenta dos factores: el salario medio de un desarrollador y el factor de overhead.

En resumen, el coste calculado por COCOMO es el coste total que le supondría a una empresa crear un software del tamaño especificado y no se ha de ver simplemente como el dinero que percibirían los programadores por realizar el software. Una vez dicho esto, los cálculos de costes dejan de parecer tan abultados.

Según estudios realizados se puede decir que en casi todos los software libres se aplica el modelo de COCOMO de forma independiente para cada una de las aplicaciones que lo integran. Tal es el caso de Debian y Red Hat. En ambos software se ha aplicado el modelo de COCOMO a cada uno de los paquetes

que lo integran para luego sumar el coste y el personal total necesario. El tiempo de desarrollo es tomado como si todos los paquetes se estuvieran desarrollando de forma paralela.

La ingeniería de Software Libre se encuentra dando sus primeros pasos. Como planes futuros está la creación de un modelo que explique como se genera el Software Libre.

Las cuestiones que se han de afrontar en un futuro próximo son: la clasificación de los proyectos de Software Libre, la creación de una metodología basada en lo posible en elementos de análisis automáticos y la utilización de los conocimientos adquiridos para la realización de modelos que permitan entender el desarrollo de Software Libre, a la vez que facilite la toma de decisiones a partir a la experiencia adquirida.

En la actualidad, tras varios años de informes y estudios de proyectos libres cuantitativos y cualitativos completos, se está realizando un enorme esfuerzo en la consecución de una infraestructura global que permita la clasificación, el análisis y la modelización de los proyectos en un espacio de tiempo limitado y de manera parcialmente automática. Cuando el análisis de los proyectos de Software Libre deje de ser tan costoso en tiempo y en esfuerzo como lo es hoy en día, es muy probable que se abra una nueva etapa en la ingeniería del Software Libre, en la que entrarán en escena otro tipo de técnicas, cuyo propósito principal se sitúe en torno a la predicción de la evolución del software y en el pronóstico de posibles complicaciones. [18]

LiveCD

Objetivos para crear un LiveCD.

Los LiveCD han tenido, y seguirán teniendo, una gran aceptación debido a que es una forma rápida, limpia y sencilla de probar cualquier distribución GNU/Linux y si es el caso instalarla. Existen muchos motivos para querer generar un sistema Live y ello depende del objetivo para el cual fue creado. Ejemplo:

- Distribuciones para objetivos específicos.
- Distribuciones para demostraciones.
- Distribuciones para instalaciones genéricas.
- Distribuciones para actividades en grupo.

- Dar facilidades a los usuarios para probar GNU/Linux.

Además de esto, se debe tener en cuenta el hecho de que el cdrom/dvd es un medio relativamente económico, mas existiendo medios regrabables. [19]

Ventajas de la creación de un LiveCD

- Son distribuciones fáciles de encontrar, ya que algunas revistas informáticas se deciden por este tipo de distribuciones para llegar al usuario de Windows.
- No hay instalación, por lo que no hay que tocar el disco duro, ni seguir procedimientos complicados. Además, los datos, particiones o sistemas operativos del disco duro no se pierden. Aún así algunas poseen un instalador para poder ser instaladas, pudiendo conocer el rendimiento real de la distribución, pues la velocidad de transferencia de las unidades lectoras (CD/DVD) es muy inferior a la de los discos duros.
- Suelen tener un reconocimiento de hardware avanzado, fruto también de las últimas versiones del kernel que suelen poseer.
- En definitiva, las distribuciones LiveCD intentan hacer llegar GNU/Linux a los usuarios de otros sistemas operativos. [20]

Requerimientos mínimos para crear un LiveCD.

Antes de encaminar el trabajo para la creación de un LiveCD se debe garantizar la existencia de algunos requerimientos mínimos, como son:

Requerimientos de hardware:

- Algunos LiveCD pueden ser ejecutados desde una Pentium II hacia arriba, con 300MHz y 96MB de RAM y 2GB en disco duro que es lo que ocuparía el sistema básico.
- Requerimientos de Software:
- Un sistema GNU/Linux con las herramientas típicas que será en la que se trabajará, por norma general será la distribución que se usa habitualmente. A esto se le llamará sistema de trabajo, es conveniente que tenga al menos 2 GB libres en la misma partición o en otra.

- Otro sistema GNU/Linux que se utilizará de base para crear el sistema Live, todo lo que instaléis aquí será lo que esté en el CD-Live. Se le llamará sistema base. La partición donde se instale es conveniente que tenga al menos 3 GB, aunque puede tener menos.
- "mkisofs" para generar la imagen final.
- Alguna herramienta para grabar los cd-roms como cdrecord en línea de comandos o k3b en modo gráfico.
- Opcionalmente también se puede instalar qemu, un emulador muy versátil al estilo de vmware pero libre, o como bosch pero más rápido, os permitirá probar la distribución live sin necesidad de reiniciar el ordenador ni tener que grabar un cdrom.
- La última versión estable del calzador [21]

Elaboración de un LiveCD

Funcionamiento interno del LiveCD

Es necesario configurar el ordenador para que arranque desde la unidad CD-ROM.

- Se reinicia el cómputo con el CD en la unidad lectora.
- El ordenador ejecuta el gestor de arranque ubicado en el LiveCD.
- Se ejecuta el Kernel y se descomprime el initrd.
- El initrd realiza los preparativos necesarios para arrancar el sistema objeto comprimido.
- Puesta en marcha del sistema objeto.
- Detección de todos los dispositivos de hardware necesarios.
- .Autoconfiguración de algunas aplicaciones como la red, la tarjeta de video y audio.
- Interacción con el sistema objeto. [**Anexo 1**]

Remasterizar un LiveCD

Remasterizar un LiveCD no es más que realizar una serie de operaciones sobre un LiveCD existente con el objetivo de aprovechar muchas de las aplicaciones de este e incluirle otras al nuevo producto.

Para crear un LiveCD es necesario seguir una serie de pasos, teniendo en cuenta ciertos requerimientos tanto de software como de hardware.

Para comenzar es importante saber que es necesario tener por lo menos 4GB de espacio libre para construir el sistema que se convertirá después en el LiveCD.

El siguiente paso es el de preparación del sistema. En dependencia del gusto y las necesidades del desarrollador se crean los directorios donde se construirá el sistema, teniendo en cuenta única y exclusivamente el espacio del que se dispone. Una propuesta interesante es crearlos en el /home o en /opt. Sources va a utilizarse para comprimir y construir en él.

```
mkdir -p livecd/sources
```

Es necesario conseguir un stage, teniendo en cuenta que si se quiere que el LiveCD pueda ser utilizado por cualquiera, se debe usar un stage genérico de x86 o i686 para que sea compatible con cualquier arquitectura.

```
cd livecd/sources
```

```
tar -xvjpf stage3-i686-2004.0
```

```
mkdir newroot
```

Después de esto se debe bajar el último portage disponible y descomprimirlo en *livecd/sources/usr*.

```
cd usr
```

```
tar -xvjpf portage-xxxx.tar.bz2
```

Con el objetivo de ganar en espacio se acude a liberar espacio en /etc. Aunque se debe tener presente que aquí hay archivos que serán necesarios en livecd/sources/etc a la hora que se haga el chroot como por ejemplo el resolv.conf para la conexión a Internet.

El paso que sigue es la instalación del sistema. Para instalar el sistema lo más importante es saberlo hacer desde el stage que se escogió. Además es necesario realizar algunos montajes para poder completar la instalación.

```
cd livecd/sources
```

```
mount -o bind /proc proc
```

```
mkdir usr/portage/distfiles
```

```
mount -o bind /usr/portage/distfiles usr/portage/distfiles/
```

Antes de proseguir con el chroot se debe verificar otros archivos importantes en livecd/sources/etc como lo es make.conf y verificar las USES oportunas para construir el sistema.

```
cd livecd/sources
```

```
chroot . /bin/bash -login
```

```
env-update
```

```
source /etc/profile
```

Después de esto se comienzan a compilar tanto los archivos necesarios como los opcionales que se deseen en el sistema.

Antes de compilar el kernel es fundamental que se configuren otros archivo del /etc; el fstab primeramente y luego otros como: (hostname, rc.conf, conf.d,...).

Compilar el squashfs-tools es realmente importante ya que ese será el sistema de archivos de la imagen comprimida del sistema que se está creando.

El siguiente paso a seguir es compilar el kernel con soporte para las diferentes arquitecturas. Hay que tener en cuenta que el kernel tiene que tener los siguientes soportes:

1. iso9660 cdrom filesystem
2. initrd support, a 8MB
3. loopback block device support
4. IDE/ATAPI cdrom device support
5. ext2 filesystem support (para ser usado en tu imagen de initrd)
6. tmpfs filesystem support

Después de analizado esto se procede a compilar el kernel.

```
emerge gentoo-dev-sources
```

```
cd /usr/src/linux
```

```
genkernel --makemenuconfig
```

```
make all && make modules_install && make install
```

Luego se hace lo mismo para grub, el gestor de arranque.

```
cd /usr/portage/sys-boot/grub/
```

```
ACCEPT_KEYWORDS="~x86" emerge grub-0.95.xxxx.ebuild
```

Ya todo está listo para crear la imagen INTRID. Este es el archivo que usará más a lo largo del arranque del LiveCD. El tamaño de este archivo va a depender de las necesidades que se tengan y será definido a través de "count". Su tamaño tiene que especificarse también en la opción del INTRID en el kernel.

```
touch /boot/initrd
```

```
dd if=/dev/zero of=/boot/initrd bs=1024k count=8
```

```
losetup /dev/loop0 /boot/initrd
```

```
mke2fs /dev/loop0
```

```
mkdir /mnt/initrd
```

```
mount /dev/loop0 /mnt/initrd
```

Después de esto se hace imprescindible la creación de los directorios y archivos necesarios para el nuevo sistema que se está creando.

```
cd /mnt/initrd
```

```
mkdir etc dev lib bin proc new cdrom
```

```
touch linuxrc
```

```
chmod +x linuxrc
```

```
touch etc/mtab
```

```
touch etc/fstab
```

Es preciso copiar los archivos necesarios dentro de bin y lib. Para bin se copiará lo siguientes archivos:

```
/bin/sh
```

```
/bin/cat
```

```
/bin/mount
```

```
/bin/mkdir
```

```
/bin/chroot
```

```
/bin/tar
```

/bin/ls

/sbin/pivot_root

Para lib hay saber que librerías son necesarias para cada uno de los binarios anteriores, esto lo haremos con ldd con cada uno de los archivos y copiar los requeridos en lib. Luego se crean los dispositivos necesarios en el directorio de dev

mknod /mnt/initrd/dev/console c 5 1

mknod /mnt/initrd/dev/null c 1 3

mknod /mnt/initrd/dev/hda b 3 0

mknod /mnt/initrd/dev/hdb b 3 64

mknod /mnt/initrd/dev/hdc b 22 0

mknod /mnt/initrd/dev/hdd b 22 64

mknod /mnt/initrd/dev/tty c 4 0

mknod /mnt/initrd/dev/loop0 b 7 0

Finalmente se necesita crear el script de linuxrc. El objetivo de este script es muy variado pues debe poder:

1. Salvar cualquier cosa que pasa por el kernel para pasar después por el /sbin/init.
2. Localizar el dispositivo del CDROM.
3. Montar el cdrom en /cdrom
4. Montar la imagen del Sistema de Archivos Zisofs en /new. Este será el sistema de archivos / con propiedades de solo lectura.

5. Crear los puntos de montaje de escritura-lectura necesarios (etc, var, tmp, y root) como tmpfs y llenarlos.

6. Finalmente el cambio de el sistema de archivos en una nueva raíz e iniciar el verdadero proceso del init.

```
#!/bin/sh
```

```
export PATH=/bin
```

```
# Adquiere kernel CMDLINE
```

```
mount -t proc none /proc
```

```
CMDLINE=`cat /proc/cmdline`
```

```
umount /proc
```

```
# Monta dispositivo de CD
```

```
CDROM=""
```

```
for x in hda hdb hdc hdd
```

```
do
```

```
mount -t iso9660 -r /dev/${x} /cdrom > /dev/null 2>&1
```

```
if [ "$?" = "0" ]
```

```
then
```

```
CDROM="${x}"
```

```
break
```

```
fi
```

```
done

# no encuentra el CD

if [ "${CDROM}" == "" ]

then

exec /bin/sh

exit

fi

# Monta root y crea los directorios de lectura-escritura

mount -t squashfs -o loop /cdrom/files/source.img /new > /dev/null 2>&1

mount -t tmpfs -o size=32m none /new/var > /dev/null 2>&1

mount -t tmpfs -o size=64m none /new/etc > /dev/null 2>&1

mount -t tmpfs -o size=10m none /new/tmp > /dev/null 2>&1

mount -t tmpfs -o size=10m none /new/root > /dev/null 2>&1

cd /new/var && tar xpf /cdrom/files/var.tar > /dev/null 2>&1

cd /new/etc && tar xpf /cdrom/files/etc.tar > /dev/null 2>&1

cd /new/root && tar xpf /cdrom/files/root.tar > /dev/null 2>&1

# Cambia root y empieza el init real

cd /new

pivot_root . newroot
```

```
exec chroot . /bin/sh <<- EOF >dev/console 2>&1
```

```
exec /sbin/init ${CMDLINE}
```

EOF

Ya está el Directorio de fuentes, pero antes de moverlo se necesita hacer algunas modificaciones. Primero en /var están directorios del portage “cache” y “db” que son muy grandes y por tanto se van a mover a /usr/lib/portage para ahorrar espacio en /var.

```
cd /var
```

```
mv cache db /usr/lib/portage/
```

```
ln -s /usr/lib/portage/cache cache
```

```
ln -s /usr/lib/portage/db db
```

Se ha terminado con el directorio de sources. Se debe salir del chroot y desmontar

```
/proc
```

y /usr/portage/distfiles, para crear el LiveCD.

Hasta este momento todas las condiciones están creadas para crear el LiveCD. Para ello se deben seguir los siguientes pasos:

1. Limpiar los directorios innecesarios como (/tmp /var/tmp).
2. Cree el directorio que utilizaremos para crear la imagen ISO (/mnt/mayix-iso).
3. Guardar los directorios de Lectura/Escritura (/etc y /var) en tar en el Directorio.
4. Usar las herramientas squasfs, para convertir el directorio de sources en una imagen squash.

5. Y finalmente crear el iso.

Para automatizar estos pasos crear un simple “build” script dentro del directorio “livecd”.

```
cd livecd
```

```
touch build
```

```
chmod +x build
```

Dicho script: es el siguiente:

```
#!/bin/bash
```

```
rm -rf target
```

```
mkdir target
```

```
cp -a source/boot target/
```

```
mkdir target/files
```

```
rm -rf source/var/tmp/*
```

```
rm -rf source/var/run/*
```

```
rm -rf source/var/lock/*
```

```
rm -rf source/tmp/*
```

```
rm -f source/etc/mtab
```

```
touch source/etc/mtab
```

```
cd source/etc/
```

```
tar cvpf ../../target/files/etc.tar * .[:alnum:]]*
```

```
cd ../var/
```

```
tar cvpf ../../target/files/var.tar * .[:alnum:]]*
```

```
cd ../root/
```

```
tar cvpf ../../target/files/root.tar * .[:alnum:]]*
```

```
cd ../../
```

```
mksquashfs source/ target/files/source.img
```

```
mkisofs -R -b boot/grub/stage2_eltorito -no-emul-boot -boot-load-size \
```

```
4 -boot-info-table -iso-level 4 -hide boot.catalog -o ~/livecd.iso target/
```

Es indispensable chequear los archivos tar que serán los que se almacenen en el tmpfs. [22].

Capítulo 3: Desarrollo e Implementación de una Distribución de GNU/Linux

Introducción

Del resultado de la Metodología y de la Ingeniería de Software surge una distribución llamada Nova BioLive.

Este capítulo va encaminado a aguantar lo concerniente al desarrollo de dicha distribución. Ofrece la arquitectura de Nova BioLive e incluye una breve descripción de las aplicaciones que van a contener cada uno de los paquetes que conforman el LiveCD.

Arquitectura de Nova BioLive

La arquitectura de Nova BioLive es el resultado de la unión entre la arquitectura de Nova y paquetes de aplicaciones con carácter puramente bioinformático. La nueva distribución heredará la mayoría de las características de su distribución base. Características que están basadas en las facilidades que brindan los componentes de la arquitectura de Gentoo y todas las flexibilidades que Nova posee; debido esencialmente a que esta distribución está basada en Gentoo.

El portage, el sistema de selección y la filosofía de desarrollo han sido asimilados y transformados en herramientas con una mejor integración con el escritorio creando front-ends que puedan ser manejados visualmente a través de GUI.

Los componentes de la arquitectura de Nova BioLive son: Kernel, Base de usuario GNU, Base de usuario Gentoo, Nova CLI, Base de usuario Gentoo, Nova CLI, Nova GUI, Nova BioLive Userland.

Kernel

El kernel y sus interfaces con el sistema son consideradas como la primera capa de componentes de Nova y actúa como base para varias de las optimizaciones y tecnologías presentes en el sistema especialmente el modelo de seguridad.

El hecho de que el kernel forme parte de la arquitectura crea una dependencia que lejos de ser dañina resulta provechosa para tareas de bechmarking y pruebas del sistema así como en el proceso de corrección de errores.

Base de usuario GNU

El proyecto GNU incluye una serie de componentes estándar importados íntegramente en Nova LNX. Estos componentes descritos por el estándar POSIX son la base de toda distribución de GNU/Linux y definen en su conjunto.

Base de usuario Gentoo

Gentoo Linux incluye un conjunto de herramientas tratadas en el capítulo 2 para el manejo básico de paquetes y configuración. Debido a que son parte fundamental del sistema y a al hecho de que alterar el funcionamiento estándar de cualquiera de ellas llevaría a la incompatibilidad parcial o total con la distribución base. La base de usuario incluye además una serie de permisos y políticas de usuarios y grupos en los que se basa la seguridad al nivel más básico.

Nova CLI

Esta es la capa responsable por interactuar directamente con cualquiera de los servicios proporcionado por cualquiera de las capas anteriores que no sea accesible fácilmente por el usuario inexperto. Consiste principalmente en scripts sustitutos o envoltorios para comandos ya existentes así como interfaces que actúen como intermediarias para algunas herramientas GUI. Esta capa incluye además los mecanismos necesarios para el control de privilegios de usuario basado en el estricto sistema de permisos que forma parte de Gentoo Linux.

Nova GUI

Siendo la fuente de interacción principal con el usuario, esta capa actúa como front-end a varios procesos de configuración y mantenimiento del sistema. Entre las herramientas actualmente desarrolladas se encuentra el instalador del sistema, software para la administración de discos duros, instalador de aplicaciones, administrador de recursos compartidos. [23]

Nova BioLive Userland

Este componente va a incluir paquetes con aplicaciones de consola y escritorio puramente bioinformáticas, siendo este el sello distintivo de Nova BioLive. [Anexo 2]

¿Cómo se creó el LiveCD de Nova BioLive?

Para la creación de Nova BioLive el primer paso a seguir fue instalar la distribución que serviría como punto de partida, posterior a esto se preparó la imagen y una vez lista, se instalaron las aplicaciones que tendría finalmente el LiveCD. A continuación una explicación detallada de todo este proceso de desarrollo:

Pasos para instalar Nova

Para hacer esta guía lo más general posible se parte de una situación inicial:

Dispositivo	Tamaño	Tipo de sistema de archivos
/dev/hda1	10GB	NTFS
—	15GB	espacio disponible

Para instalar el sistema GNU/Linux se va a aprovechar el espacio disponible de 15GB.

1. Particionar:

Debemos crear varias particiones, las mínimas recomendadas son:

- Una partición SWAP que nos servirá como expansión de memoria en caso de necesitar un poco más de la disponible.
- En nuestro caso de 256GB.
- Una partición para el sistema que debe tener como mínimo recomendable 5GB (se recomiendan 10GB).
- En nuestro caso de 8GB.
- Una partición para /home donde los usuarios podrán copiar sus archivos, para esta utilizaremos el resto del espacio disponible.
- Aquí aprovecharemos todo el espacio disponible.

2. Dar formato a las particiones:

Al crearlas en ese orden quedarían para la swap la /dev/hda2, para el sistema /dev/hda3 y para /home la /dev/hda4 por lo que se ejecutan los siguientes comandos:

```
# mkswap /dev/hda2 – Dar formato de SWAP a /dev/hda2
```

```
# swapon /dev/hda2 – Comenzar a utilizar la swap
```

```
# mkreiserfs /dev/hda3 – Dar formato de REISER a /dev/hda3
```

```
# mkreiserfs /dev/hda4 – Dar formato de REISER a /dev/hda4
```

3. Montar las particiones que utilizaremos:

```
# mount /dev/hda3 /mnt/gentoo – Montar /dev/hda3 en /mnt/gentoo para acceder a los datos
```

`# mkdir /mnt/gentoo/home` – Crear el directorio /home en la partición /dev/hda3 para que al descomprimir se copie hacia ahí el contenido

`# mount /dev/hda4 /mnt/gentoo/home` – Montar /dev/hda4 en el directorio home de la partición para el sistema.

4. Copiarel archivo con la imagen del sistema a /mnt/gentoo.

5. Descomprimimos el archivo de imagen:

`# cd /mnt/gentoo` – entrar al directorio donde montar la partición para el sistema

`# tar xjpf Imagen.tar.bz2` – se descomprime la Imagen estando ya en el directorio.

6. Configurar el sistema según las condiciones de la PC:

(estando en el directorio /mnt/gentoo)

- Editar el archivo etc/fstab

`# nano etc/fstab`

y lo ajustar a las particiones quedando el archivo como:

```
/dev/hda2  none      swap      sw                0      0
/dev/hda3  /          reiserfs  noatime,acl,user xattr  0      1
/dev/hda4  /home     reiserfs  defaults,noatime,acl,user xattr  0      1
proc       /proc     proc      defaults          0      0
shm        /dev/shm  tmpfs     nodev,nosuid,noexec  0      0
```

- Editar el archivo boot/grub/grub.conf

`# nano boot/grub/grub.conf.`

y lo ajustar a las particiones quedando el archivo como:

```
default=0
```

```
timeout=5
```

```
splashimage=(hd0,2)/boot/grub/splash.xpm.gz
```

```
title Nova
```

```
root (hd0,2)
```

```
kernel (hd0,2)/boot/kernel-genkernel-x86-2.6.19-gentoo-r5 root=/dev/ram0 real root=/dev/sda5 elevator=cfq  
video=vesafb:
```

```
1024x768-32@65 splash=silent,fadein,theme:nova console=tty1
```

```
initrd (hd0,2)/boot/initramfs-genkernel-x86-2.6.19-gentoo-r5
```

7. Pasar a instalar el cargador del sistema, en nuestro caso grub.

Montar los dispositivos necesarios para acceder al Disco Duro:

```
# mount -t proc none proc/
```

```
# mount -o bind /dev/ dev/
```

y cambiar nuestro sistema por el que instalar

```
# chroot .
```

```
# grub – Accedemos al grub
```

```
root (hd0,2) – Se le dice al grub dónde está su archivo de configuración
```

```
setup (hd0) – Se le dice que se instale al inicio del Disco Duro
```


quit – Se sale del grub.

8. Le ponemos una clave al usuario root (estando aun en el sistema que instalar)

```
# passwd
```

9. Salir del sistema y desmontar las particiones

```
# exit
```

```
# cd ..
```

```
# umount gentoo/proc/ gentoo/dev/ gentoo/
```

10. Reiniciare iniciar el sistema.

Preparación de la imagen

Se necesita tener por lo menos 6GB de espacio libre para construir nuestro sistema que se convertirá después en el LiveCD.

- Se deben crear las carpetas donde se realizará el LiveCD:

```
mkdir /target/files/source (En source es donde se va a descomprimir la imagen del LiveCD)
```

- Luego de descomprimir la imagen hacemos algunos montajes para poder seguir con la instalación.

```
cd /target/files/source
```

```
mount -o bind /proc proc
```

```
mkdir usr/portage/distfiles
```

```
mount -o bind /usr/portage/distfiles usr/portage/distfiles/
```

- Debemos verificar otros archivos importantes (antes de proseguir con el chroot) en

/target/files/source/etc como lo es *make.conf* y verificar las USES oportunas para construir nuestro sistema como por ejemplo *USE="kerberos bash-completion seamonkey -qt3 -qt4 -arts wma -kde -oss nfs"*.

- Luego se pasa al chroot para la posterior compilación del kernel:

```
cd /target/files/source
```

```
chroot . /bin/bash -login
```

```
env-update
```

```
source /etc/profile
```

Vale aclarar que se debe tener instalado en el sistema que se va a hacer el LiveCD el paquete *squashfs-tools* para la compresión del sistema LiveCD.

Sería bien importante que se tomara en cuenta el paquete de *livecd-tools* que será de mucha ayuda cuando el sistema arranque desde el CD.

- El siguiente paso es que compilemos el kernel con soporte para las diferentes arquitecturas para lo cual se deben tener en cuenta los siguientes soportes:

```
iso9660 cdrom filesystem
```

```
initrd support, a 8MB
```

```
loopback block device support
```

```
IDE/ATAPI cdrom device support
```

```
tmpfs filesystem support
```

- Ahora se va a compilar el kernel:

```
emerge gentoo-dev-sources
```

```
cd /usr/src/linux
```

```
genkernel --makemenuconfig
```

- Ahora se pasa a configurar el fstab, que se localiza en: /etc/ y el menu.lst que se encuentra como vínculo al grub.conf en /boot/grub/:

(Se debe eliminar el link menu.lst y crear un archivo con el mismo nombre y copiar la configuración)

```
/dev/loop0    /          squashfs    ro,defaults    00
none          /proc      proc        defaults        00
none          /dev/shm   tmpfs       defaults        00
none          /dev/pts   devpts      defaults        00
```

(Si la imagen no tiene grub compilado, simplemente se ejecuta el comando en consola: emerge grub)

Aquí está el /boot/grub/grub.conf

```
default 0
```

```
timeout 5
```

```
splashimage=(cd)/boot/grub/splash.xpm.gz
```

```
title=Gentoo Linux
```

```
root (cd)
```

```
kernel (cd)/boot/kernel-xxxx video=vesafb:1024x768-32 root=/dev/ram0 rw      init=/linuxrc cdroot
```

```
initrd (cd)/boot/initrd
```

- Otro de los aspectos importantes es la configuración de la sesión. Aquí se tiene que elegir un fondo adecuado y un tema de la pantalla de inicio de sesión del entorno de escritorio de Linux GNOME (GDM).

Instalación de las aplicaciones

La mayoría de las aplicaciones seleccionadas se ejecutan solo por consola por lo que para facilitar el trabajo con ellas se le agrega al panel una carpeta con todas las aplicaciones separadas por paquetes de la siguiente manera:

En *System/Preference/Menu Layout* se agrega una nueva carpeta (New Menu) y dentro de ella se va creando con la opción New Item todos los links a las aplicaciones que ya se tienen instaladas. Esto es solo para las aplicaciones que tienen interfaz gráfica.

Otro detalle para poder correr las aplicaciones es que se necesita aceleración gráfica o direct rendering, esto se debe compilar en el kernel y agregarlo al `xorg.conf` que se encuentra en `/etc/X11/`. Esto último es en caso que no se tenga aceleración gráfica en la imagen del LiveCD, sino no es necesario. Para comprobar esto se puede ejecutar el comando en consola `glxinfo` y chequear en la línea del direct rendering si está en Yes.

Creación del LiveCD

- Para la creación del LiveCD se necesitan hacer algunas modificaciones:

Primeramente en `/var` están directorios del portage “cache” y “db” que son muy grandes así que los se va a mover a `/usr/lib/portage` para ahorrar espacio en `/var`. Esto se debe hacer desde un chroot al sistema del liveCD.

```
cd /var
```

```
mv cache db /usr/lib/portage/
```

```
ln -s /usr/lib/portage/cache cache
```

```
ln -s /usr/lib/portage/db db
```

Otro archivo que se debe mover es `/etc/gconf/` porque debemos tratar que la carpeta `/etc` nos quede lo más chiquita posible ya que como en esta carpeta se encuentran archivos que sufren modificaciones durante el uso del liveCD como tal, la máquina le asigna una cantidad de memoria y si excede esta cantidad el liveCD simplemente no levantará.

```
cd /etc
```

```
mv gconf /usr/lib/portage/
```

```
ln -s /usr/lib/portage/gconf gconf
```

También es necesario mover o eliminar otros archivos que no son necesarios en el liveCD. Los que están especificados como Mover, simplemente pueden ser salvados en una carpeta fuera de la carpeta `/target`.

```
var/tmp/*
```

(Eliminar)

```
var/cache/*
```

(Mover)

```
var/db/*
```

(Mover)

```
usr/portage
```

(Mover)

usr/share/doc

(Mover)

usr/src/*

(Mover)

var/log/*

(Eliminar)

etc/portage/*

(Mover)

etc/make.conf

(Mover)

etc/fstab

(Mover)

etc/hosts

(Mover)

etc/resolv.conf

(Mover)

etc/conf.d/net

(Mover)

root/*

(Eliminar)

tmp/*

(Eliminar)

mnt/*

(Eliminar)

.keep

(Eliminar)

Las carpetas que dicen "Mover" las copio fuera de la carpeta /target en otra carpeta, pues no me conviene eliminarlas de la computadora. Pues por ejemplo en var/db/* es donde se guardan los paquetes instalados y si después que yo tenga mi liveCD hecho me doy cuenta de que necesito instalar algo mas seria un gran problema si ya he borrado esta carpeta.

- Posterior a estas modificaciones lo primero que hay que hacer es poner el script menulive (es importante que se ponga con el mismo nombre a la hora de crearlo y además debe ser creado en la misma dirección donde se encuentra el directorio target). Este es el script:

```
#!/bin/bash +x
```

```
[[ -f functions/funkts ]] && source functions/funkts
```

```
echo "-----"
```

```
echo "La direccion de trabajo es $LIVECD"
```

```
echo "-----"
```

echo "1 para hacer la imagen livecd.squashfs y el livecd.iso"

echo

echo "2 para hacer solo el livecd.iso"

echo

echo "presione ctrl+c para salir"

echo

read -p "entre una de las opciones: " A

echo "-----"

if [-n "\$A"]; then

cd scripts/

pwd

if [\$A = "1"]; then

build

elif [\$A = "2"]; then

iso

else

echo "\$A es una opcion invalida"

#cd ..


```
. menulive  
  
fi  
  
cd ..  
  
fi
```

Este script me redirecciona para otro script llamado `funkts` que debe estar en una carpeta llamada `functions` y que debe estar creada en la misma dirección donde fue creada `/target`.

- Luego corremos el script que lleva por nombre "funkts":

```
#!/bin/bash  
  
BUILD='/mnt/livecd/target' #Ubicacion de la carpeta target.  
  
KERNEL=kernel-genkernel-x86-2.6.20-gentoo-r8 #Nombre del kernel que se está utilizando en la imagen  
del LiveCD.  
  
INITRD=initramfs-genkernel-x86-2.6.20-gentoo-r8 #Nombre del initrd  
  
LIVECD='/mnt/livecd' #Ubicacion de la carpeta donde se está construyendo el LiveCD.  
  
build() {  
  
    cd $LIVECD/target/files  
  
    rm $LIVECD/target/livecd.squashfs  
  
    time mksquashfs source/ $LIVECD/target/livecd.squashfs # -no-fragments  
  
    touch $LIVECD/target/livecd # !!! This file must be on livecd root, because init script in  
    initramfs uses this file to identify if cd is mounted or no!!!
```

```
iso    }

iso() {

    CDLABEL="NovaBioLive"

    copyboot

    cd $LIVECD

    mkdir -p temp

    mv $LIVECD/target/files $LIVECD/temp

    cd $LIVECD/target/

    time mkisofs -V $CDLAVEL -R -b boot/grub/stage2_eltorito -no-emul-boot -boot-load-size 4 -boot-
info-table -iso-level 4 -hide-rr-moved -c boot.catalog -o novabiolive.iso $LIVECD/target/

    mv $LIVECD/temp/files $LIVECD/target/

    }

kernelchangenamename() {

    mv $LIVECD/target/boot/$INITRD $LIVECD/target/boot/initrd

    mv $LIVECD/target/boot/$KERNEL $LIVECD/target/boot/vmlinuz

    }

copyboot() {

    echo " >> copiando boot"
```

```
cp -a $LIVECD/target/files/source/boot/ $LIVECD/target/  
  
kernelchangenname  
  
}
```

Este script en general se encarga de crear el liveCD con una función llamada "mkisofs".

Características propias de Nova BioLive

Paquetes de Nova BioLive

Para el desarrollo de esta meta distribución se seleccionaron paquetes que son muy populares dentro del ámbito de la bioinformática a nivel mundial y que además abarcan diferentes ramas de esta ciencia como la Biología Molecular, Química Teórica y Biología de Sistemas.

A continuación se muestran las aplicaciones que se seleccionaron para esta edición del LiveCD y las funcionalidades de las mismas:

Biología Molecular

Bioperl: Esta aplicación se ha creado con el objetivo de formar programadores en lenguaje Perl que puedan manipular secuencias de ADN (eventualmente tomadas de bases de datos públicas) con el fin de extraer información sobre genes, simular mutaciones, etc. [24]

ClustalX: Es una versión del programa múltiple de alineación de secuencia de Clustal W con un interfaz gráfico. Mantiene un ambiente integrado realizando sucesión múltiple, alineaciones del perfil y análisis de los resultados. [25]

ClustalW: Los alineamientos múltiples de secuencia de proteínas son las herramientas importantes en el estudio de las secuencias. La información básica que ellos proporcionan es identificación de regiones de las secuencias conservadas. Esto es muy útil en los experimentos diseñados para probar y modificar la

función de proteínas específicas, prediciendo la función y estructura de proteínas, e identificando a los nuevos miembros de la familia de las proteínas. [26]

EMBOSS tools: EMBOSS (Suite Europea de Software Libre de Biología Molecular) es una colección de herramientas de bioinformática del software libre para el análisis de secuencia. Hay una amplia variedad de programas que componen la SUITE, extendiéndose en aplicaciones desde búsqueda de Base de Datos hasta presentación de secuencias de datos. Es sumamente simple instalar este software, configurarlo y amigable a la hora de ser usado. [27]

Glimmer: Es un sistema utilizado para analizar el ADN y de esta manera encontrar los genes, especialmente el genoma de bacterias y los virus. [28]

GROMACS: Es originalmente un paquete de simulación dinámica molecular desarrollado en la Universidad de Groningen. Este proyecto desde sus inicios se hizo con el objetivo de construir un sistema paralelo especializado para las simulaciones moleculares. [29]

NCBI tools: Es un juego de herramientas para usar con más eficiencia NCBI, estas herramientas se dividen en tres partes principales: Modelo de Datos, Codificación de Datos, Programa de Bibliotecas. [30]

NJPlot: Es un programa de dibujo de “árboles” capaz de dibujar cualquier árbol filogenético expresado en el formato de árbol filogenético Newick (Ej, el formato usado por el paquete PHYLIP). NJplot es esencialmente conveniente para enraizar los árboles que no tienen raíz obtenidos desde la parsimonia (severidad, discreción, prudencia), distancia o métodos de construcción de árboles de máxima probabilidad. [31]

PyMOL: Es un sistema de visualización molecular. Permite la visualización en 3D de proteínas, moléculas pequeñas, densidad, superficies, y trayectorias. [32]

Raster3D: Genera imágenes de proteínas de alta calidad y de otras moléculas. [33]

SeaView: Es un editor gráfico de alineación de sucesión múltiple. [34]

T-Coffee: Es un editor gráfico de alineación de sucesión múltiple. Esta aplicación permite combinar resultados obtenidos con varios métodos de alineación. Por ejemplo si se tiene una alineación que viene de ClustalW y otra de Dialign, y una alineación estructural de algunas de sus sucesiones, el T-Coffee combinará toda esa información y producirá una nueva sucesión múltiple. [35]

Química Teórica

Gchemical: Es un paquete de química computacional autorizado bajo el GNU GPL. [36]

MPQC: Computa propiedades de átomos y moléculas de primeros principios usando la ecuación de Schrödinger. Corre en una gama amplia de arquitecturas que van de los puestos de trabajo individuales a los multiprocesadores simétricos a las computadoras masivamente paralelas. [37]

Open Babel: Es una caja de herramientas química diseñada para hablar los muchos idiomas de datos químicos. Es un proyecto abierto, colaborativo que permite investigar, convertir y analizar datos que tienen que ver específicamente con las áreas relacionadas a la química y la bioquímica. [38]

PSI3: Es un paquete de química computacional autorizado bajo el GNU GPL

Rasmol: Es un programa de gráficos moleculares que permite la visualización de cualquier tipo de estructura molecular definida. Además de que permite colorear, resaltar y seleccionar átomos y/o regiones particulares, por lo que facilita el aprendizaje de los fenómenos estructurales y de su relación con la actividad biológica. [39]

XdrawChem: Es una aplicación diseñada para dibujar y analizar estructuras químicas y reacciones para los sistemas operativos de Unix. [40]

TINKER: Esta aplicación está diseñada para ser usada fácilmente y es un sistema flexible de programas y rutinas para los mecanismos moleculares. Además de que proporciona un juego de programas relativamente pequeños para realizar los cálculos complejos.

Biología de Sistemas

BioUML: Es una plataforma en JAVA para Biología de Sistemas. Incluye acceso a Bases de Datos con datos experimentales, herramientas para descripción formalizada de estructura y funcionamiento de sistemas biológicos así como herramientas para su simulación y visualización. [41]

BioGrid: Grid, un sistema en desarrollo imprescindible para tareas como el análisis de los datos que generarán los detectores del nuevo gran acelerador de partículas LHC (del Laboratorio Europeo de Física de Partículas, CERN), El Grid conjuga las bases de datos distribuidas y la computación distribuida, con líneas de intercomunicación de alta velocidad. La idea es poder utilizar centenares de ordenadores diferentes dispersos en institutos y centros de muchos países para una misma tarea de cómputo, de forma que cada uno almacena y computa una parte del cálculo. Su versión biológica es el Biogrid. [42]

CellDesigner: Es un programa de uso libre para la estandarización de la representación de vías y procesos a nivel molecular en el lenguaje SBML, específicamente para diseñar proteínas y otros elementos de las redes celulares. [43]

CellML2SBML: CellML y SBML son lenguajes basados en XML para almacenamiento e intercambio de modelos biológicos. Ambos utilizan un subconjunto similar de la especificación MathML para describir los aspectos matemáticos de los modelos. Para convertir un modelo válido en otro se crean 4 hojas de estilo XSLT que se aplican consecutivamente. Los modelos se convierten sin pérdida de información utilizando los estándares.

Cellware: Es una herramienta de modelación y simulación basada en grid. Es un software multiplataforma desarrollado en Java. Importa y exporta modelos con formato SBML.

Cytoscape: Es un paquete de software diseñado para posibilitar la visualización y análisis de redes. Aunque se usa comúnmente para análisis de redes biológicas, también se utiliza para ajustar nodos en grafos.

SBML Editor: Es una aplicación de biología de sistemas que facilita la creación rápida y la edición de SBML. Los usuarios pueden crear y pueden quitar lo que desean mediante este software a SBML de una manera controlada que mantiene la validez del último archivo de SBML.

SBW: Es una plataforma y colección de herramientas proporcionando capacidad de análisis, modelación y simulación. Los desarrolladores pueden utilizar estas herramientas como una reutilización del código sin la necesidad de reescribirlas para proveer funcionalidades existentes lo que permite enfocarse solamente a las nuevas tareas. Una aplicación escrita en cualquier lenguaje de programación se puede modificar para que interactúe con SBW lo que posibilita usar su funcionalidad.

¿Por qué se escogieron estas aplicaciones?

La Bioinformática, como se ha explicado en capítulos anteriores, es una ciencia multidisciplinaria. Esto significa que intervienen, en las investigaciones y desarrollos en Bioinformática, especialistas de diferentes áreas de la ciencia e incluso de la tecnología. Cada uno de estos especialistas utiliza herramientas computacionales diferentes para atacar los problemas que en su campo se presenten. De esta forma es muy común encontrar, en un laboratorio de Bioinformática, una variedad de aplicaciones incluso muchas de ellas para alcanzar objetivos similares.

Lograr una integración coherente de estas aplicaciones es de por sí un trabajo complicado y algo en lo que se puede trabajar con resultados tan importantes como necesarios. En la actualidad un especialista en Genómica utiliza una variedad de herramientas computacionales básicamente por la formación que puede haber obtenido y la experiencia en la utilización de las mismas. Estas herramientas pueden ser completamente diferentes a las que utilizaría probablemente otro especialista, también en Genómica, para alcanzar los mismos objetivos u objetivos similares. La diversidad de aplicaciones también puede depender, además de la experiencia en las mismas, de la plataforma computacional (tanto de software como de hardware) que utilice el especialista.

Es por ello que es prácticamente imposible decir “.. esta es la única herramienta que puede utilizar un biólogo para hacer el alineamiento de una familia de proteínas...”. Si se acepta esta afirmación como cierta, se estaría negando la diversidad que, en materia de Software Libre, es uno de los pilares fundamentales en que se sustenta.

Considerando esto se decide no desgastar el trabajo en una selección rigurosa de las aplicaciones que incluiría el LiveCD que proponemos, puesto que cualquier selección previa que se haga antes de la elaboración del mismo tendrá tantos adeptos como detractores pues es imposible hacer converger en una sola distribución ligera que se distribuye como LiveCD, todo el cúmulo y la variedad de aplicaciones que para un mismo tema utilizan los especialistas.

Se seleccionaron pues algunas que pueden resultar representativas para un determinado fin y que son muy populares por los criterios que se recogieron de los expertos en la temática. Otras que también aparecen en el LiveCD lo están por el hecho de sus prestaciones multiplataforma. De igual forma aparecen algunas que son de utilidad para los informáticos, etc. Cada una de estas aplicaciones que se incluyeron tiene una gran cantidad de otras aplicaciones similares pero que no pueden incluirse también porque perdería el sentido de la ligereza del LiveCD que fue el objetivo que se propuso inicialmente.

Si se piensa en un repositorio de software, se llega a la conclusión de que las ventajas de esta filosofía de trabajo permite que el usuario no recargue el entorno de desarrollo, o de trabajo, con diversidad de aplicaciones para el mismo fin, sino que con tener acceso a un lugar desde donde se pueda instalar cualquier aplicación que se necesite, lo pueda hacer sin esfuerzo alguno, dejando sano el entorno y libre de aplicaciones redundantes.

De esta forma se puso en el LiveCD un número mínimo de aplicaciones básicas que son comunes a muchos especialistas, con la posibilidad de crecer en la configuración tanto como sea deseable y necesario. No quiere esto decir que el hecho de que en el LiveCD aparezcan algunas aplicaciones, éstas sean las únicas que se pueden utilizar porque de ser así se estaría perdiendo el sentido no solo de éste trabajo sino incluso, como se dijo anteriormente, de la filosofía del Software Libre.

Por otra parte esto nos da la posibilidad incluso de personalizar mucho más este resultado en el sentido de que, de ser necesario, pudiera distribuirse un LiveCD solamente dedicado a las investigaciones en Genómica o en Proteómica, por solo poner dos ejemplos, y en cualquier caso construir el LiveCD con un número elevado de aplicaciones del mismo tipo pero especializadas en una misma área.

La riqueza de este trabajo consiste, entre otras cosas, en poder contar con este resultado y de manera muy flexible, rápida y dinámica, tener un producto de calidad y con una utilidad práctica incuestionable.

Calidad

Debido a que ninguna de los standards de calidad es aplicable al desarrollo de NOVA, se pretende utilizar un modelo propio, derivado del tradicional en los proyectos de SWL que garantiza un proceso de mejora continua en las aplicaciones y versiones de la distribución.

Estaría basado en pruebas de aceptación y gestionado por un sistema de reporte de fallas, la idea es distribuir varias versiones del producto y controlar los errores encontrados, para corregirlos en la próxima versión.

Versión Alpha:

Versión en fase experimental pero funcional. Se reportan todos los errores encontrados.

Versión Beta:

Versión en fase de pruebas, derivada de la Alpha. Corregidos todos los errores encontrados en la versión anterior. Se reportan todos los errores encontrados.

Candidato a Liberación:

Versión que ya superó la fase de pruebas y dos iteraciones de control de errores. Corregidos todos los errores encontrados en la Versión Beta. Se reportan todos los errores encontrados (deben ser muy escasos).

Versión Liberada:

Versión que ya superó tres iteraciones de control de errores. Corregidos todos los errores encontrados en la versión anterior. Se reportan todos los errores encontrados (deben ser muy escasos). Se elaboran parches a los errores encontrados.

Conclusiones

El estudio acerca de las principales distribuciones de Linux personalizadas para la bioinformática existentes en el mundo, llevo a la creación de una nueva distribución en formato LiveCD.

Nova BioLive es una distribución basada en Nova que aprovecha todas las facilidades de Gentoo e incluye un grupo de las aplicaciones mas utilizadas en el campo de la Bioinformática.

El principal aporte de Nova BioLive es que pone en manos de los biólogos herramientas que permiten el trabajo en las diferentes áreas de la Bioinformática como la Biología Molecular, la Química Teórica, la Biología de Sistemas y el Diseño de Fármacos; además es una distribución fácil de manejar y debido a su condición de LiveCD no obliga al usuario a tener instalado GNU/Linux.

Referencias Bibliográficas

- [1] Lic. Valdivia Joan Alain y Dr. Rodríguez Febles Pedro Juan C. *Bioinformática: reflexiones y perspectivas*. [en línea], 28/12/2005. Wikilearning. Disponible en: http://www.wikilearning.com/desarrollo_de_la_bioinformatica_en_cuba-wkccp-8037-2.htm. [Consulta: 13-9-2006].
- [2] Msc. Ramírez Carlos Juan. *Bioinformática en Cuba*. [en línea], La bioinformática en la obtención de vacunas. Disponible en: <http://www.bvv.sld.cu/download.php?url=117072029420.pdf> [Consulta: 13-9-2006].
- [3] [en línea], 28/12/2005. Centro de Inmunología Molecular. Disponible en: <http://www.cim.sld.cu/> . [Consulta: 13-9-2006].
- [4] Dr. Otero Glen. *Tacc Seminar Series: BioBrew, A Quaff for Life Scientists*. [en línea], 03/19/2004. Texas Advanced Computing Center. The University of Texas at Austin. Disponible en: http://www.tacc.utexas.edu/general/news/archive/20040311_01.php. [Consulta: 18-10-2006].
- [5] Rodríguez M. Carlos. *Bioknoppix was developed*. . [en línea], 2004-06-22. Disponible en: <http://bioknoppix.hpcf.upr.edu/>. [Consulta: 18-10-2006].
- [6] Biolinux Grupo. *SaluX: El Proyecto*. [en línea], 19-9-2006. Disponible en: <http://salux.org/blog>. [Consulta: 18-10-2006].
- [7] Biolinux Grupo. *SaluX: Detalles Técnicos*. Disponible en: http://salux.org/blog/?page_id=7 [Consulta: 18-10-2006].
- [8] Calejero A. Jesús. *DNALinux*. [en línea], 8-1-2006. *La Paradoja de San Petersburgo*. Disponible en: <http://calejero.blogspot.com/2006/01/dnalinix.html>. [Consulta: 20-10-2006].
- [9] Pinzón M. Andrés. *Linux Hélice*. [en línea], 2004-2006. Disponible en: <http://www.andrespinzon.com/helice.php> [Consulta: 20-10-2006].

- [10] [en línea], 2007. *Debian*. Disponible en: <http://www.debian.org/index.es.html> [Consulta: 25-3-2007].
- [11] *WDebian-Wikipedia, la enciclopedia libre*. [en línea], 12-3-2007. Wikipedia, la enciclopedia libre Disponible en: <http://es.wikipedia.org/wiki/Debian> [Consulta: 25-3-2007].
- [12] *WMandriva Linux-Wikipedia, la enciclopedia libre*. [en línea], 1-4-2007. Wikipedia, la enciclopedia libre. Disponible en: http://es.wikipedia.org/wiki/Mandriva_Linux [Consulta: 2-4-2007].
- [13] *Ubuntu (distribución Linux)*. [en línea], 19-3-2007. Wikipedia, la enciclopedia libre Disponible en: [http://es.wikipedia.org/wiki/Ubuntu_\(distribuci%C3%B3n_Linux\)](http://es.wikipedia.org/wiki/Ubuntu_(distribuci%C3%B3n_Linux)) [Consulta: 25-3-2007].
- [14] *Suse Linux*. [en línea], 25-3-2007. Wikipedia, la enciclopedia libre. Disponible en: http://es.wikipedia.org/wiki/SUSE_Linux. [Consulta: 25-3-2007].
- [15] *RedHat Linux* [en línea], 8-3-2007. Wikipedia, la enciclopedia libre. Disponible en: http://gl.wikipedia.org/wiki/Red_Hat_Linux. [Consulta: 25-3-2007].
- [16] *Gentoo Linux* [en línea], 24-3-2007. Wikipedia, la enciclopedia libre. Disponible en: <http://es.wikipedia.org/wiki/Gentoo> . [Consulta: 25-3-2007].
- [17] *Mi Distro, Gentoo*. [en línea], 22-3-2007. Donde no llegaron las hadas. Disponible en: <http://elchicosinhada.wordpress.com/2007/03/22/mi-distro-gentoo/>. [Consulta: 18-5-2007].
- [18] Barahona González Jesús, Pascual Soano Joaquín, Robles Gregorio. *Software Libre. Introducción al Software Libre*. [en línea], 21-9-2003. Disponible en: [http://unicornio.freens.org/profpcm-
aux/sobre/IntroSoftLibreUOC.pdf](http://unicornio.freens.org/profpcm-
aux/sobre/IntroSoftLibreUOC.pdf). [Consulta: 17-5-2007].
- [19] García Espino Jesús. *Generación de distribuciones live con metadistro*. [en línea], 26-4-2006. Disponible en http://www.gpul.org/jornadas06/charlas_jornadas06/docs/2006_JSL_10_metadistros.beamer.pdf. [Consulta: 7-4-2007].

- [20] *LiveCd*. [en línea], 25-3-2007. Wikipedia, la enciclopedia libre. Disponible en: http://es.wikipedia.org/wiki/Live_CD . [Consulta: 7-4-2007].
- [21] *Creación de un CD-Live con metadistros*. Badopi. [en línea], 27-5-2004. Barcelona GUSL. Disponible en: http://www.badopi.org/cdlive_metadistros. [Consulta: 7-4-2007].
- [22]. Abate Antonio Guillermo. *Creación de Live-cd Alpha-0.2* Disponible en: <http://www.mayix.net/dmdocuments/Creacion%20de%20livecd-alpha2.pdf> [Consulta: 7-4-2007].
- [23] González Herrera Anielkis y Martines Rodríguez Yoandy. *Desarrollo y Mantenimiento De Una Distribución de Linux*. 2006. [Consulta: 7-4-2007].
- [24] *BioPerl* [en línea], 14 June 2007. Wikipedia, la enciclopedia libre. Disponible en: http://www.bioperl.org/wiki/Main_Page [Consulta: 16-6-2007].
- [25] *Clustal X* [en línea], 08-7-2007. Disponible en: <http://bips.ustrasbg.fr/fr/Documentation/ClustalX/> [Consulta: 17-06-2007]
- [26] *Clustal W* [en línea], 8-1-2006. Disponible en: <http://www.ebi.ac.uk/clustalw/>[Consulta: 17-06-2007].
- [27] *EMBOSS tolos* [en línea], 8-1-2006. Disponible en: <http://bioinfo.cnio.es/EMBOSS/gui/> [Consulta: 17-06-2007].
- [28] A.L. Delcher, K.A. Bratke, E.C. Powers, and S.L. Salzberg. [Identifying bacterial genes and endosymbiont DNA with Glimmer](#). *Bioinformatics* [en línea], 2007. Disponible en: <http://www.cbcu.umd.edu/software/glimmer/> [Consulta: 17-06-2007].
- [29] *GROMACS* [en línea], 25-5-2007. Wikipedia, la enciclopedia libre. Disponible en: <http://en.wikipedia.org/wiki/GROMACS> [Consulta: 16-6-2007].
- [30] En la Web. *NCBI tolos. Herramientas del NCBI*. [en línea], 30-6-2004. Centro Nacional de Cálculo Científico. Universidad de los Andes. Disponible en:

http://www.cecalc.ula.ve/recursos/software/fichasoftware/parser.php?URL=NBCI_tools.xml&XSL=software.xsl. [Consulta: 17-06-2007].

[31] *NJplot* [en línea], 2007. Disponible en: <http://pbil.univ-yon1.fr/software/njplot.html> [Consulta: 17-06-2007].

[32] *PyMOL Molecular Graphics System*[en línea], 2007. Disponible en: <http://sourceforge.net/projects/pymol/> [Consulta: 17-06-2007].

[33] *Biomolecular Structure Center at the University of Washington. Raster3d*. [en línea], 2007. Disponible en: <http://skuld.bmsc.washington.edu/raster3d/> [Consulta: 17-06-2007].

[34] *SeaView* [en línea], 2007. Disponible en: <http://pbil.univ-lyon1.fr/software/seaview.html> [Consulta: 17-06-2007].

[35] *T-Coffee*[en línea], 2007. Disponible en: <http://www.ebi.ac.uk/t-coffee/> [Consulta: 17-06-2007].

[36] *Ghemical*. [en línea], 12-04-2006. Disponible en: <http://www.uku.fi/~thassine/projects/gchemical/> [Consulta: 17-06-2007].

[37] *The Massively Parallel Quantum Chemistry Program. MPQC*. [en línea], 2007. Disponible en: <http://www.mpqc.org/>[Consulta: 17-06-2007].

[38] *SourceForge.net. Open Babel*. [en línea], 2007. Disponible en: <http://sourceforge.net/projects/openbabel/> Consulta: 17-06-2007].

[39] *Academia Colombiana de Ciencias Exactas, Físicas y Naturales*. Disponible en: <http://www.accefyn.org.co/rasmol/informacion.html#queson> [Consulta 16-6-2007]

[40] *XDrawChem: Molecule structure drawing. XDrawChem*. [en línea], 30-11-2005. Disponible en: <http://xdrawchem.sourceforge.net/> [Consulta: 17-06-2007].

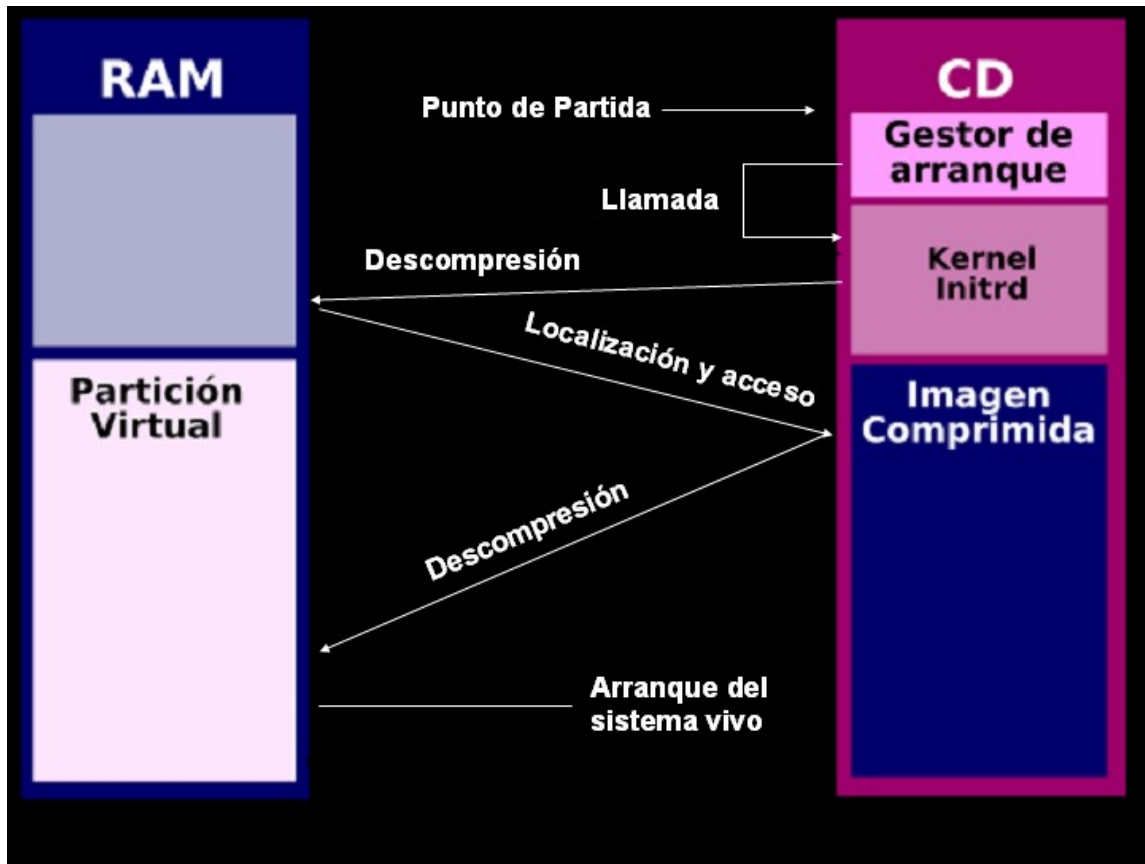
[41] En la Web. *Whats is BioUML*. BioUML - open source Java framework for systems biology. [en línea], 2001-2007. Disponible en: <http://www.biouml.org/> [Consulta 16-6-2007].

[42] Superordenadores para descifrar la vida. *La biología exige elevada potencia de cómputo para los grandes flujos de datos que genera*. [en línea], 26/02/2003 Disponible en: http://www.elpais.com/articulo/futuro/Superordenadores/descifrar/vida/elpepusocfut/20030226elpepifut_1/Tes. [Consulta 16-6-2007].

[43] Castaño G. Jose. *Las nuevas titulaciones de grado de bioquímica y biotecnología*. [en línea], marzo/2006. Disponible en: <http://www.sebbm.com/pdf/147/d04147.pdf> [Consulta 16-6-2007]

Anexos

Anexo 1: Esquema que muestra los saltos que hace la computadora para arrancar un CD vivo.

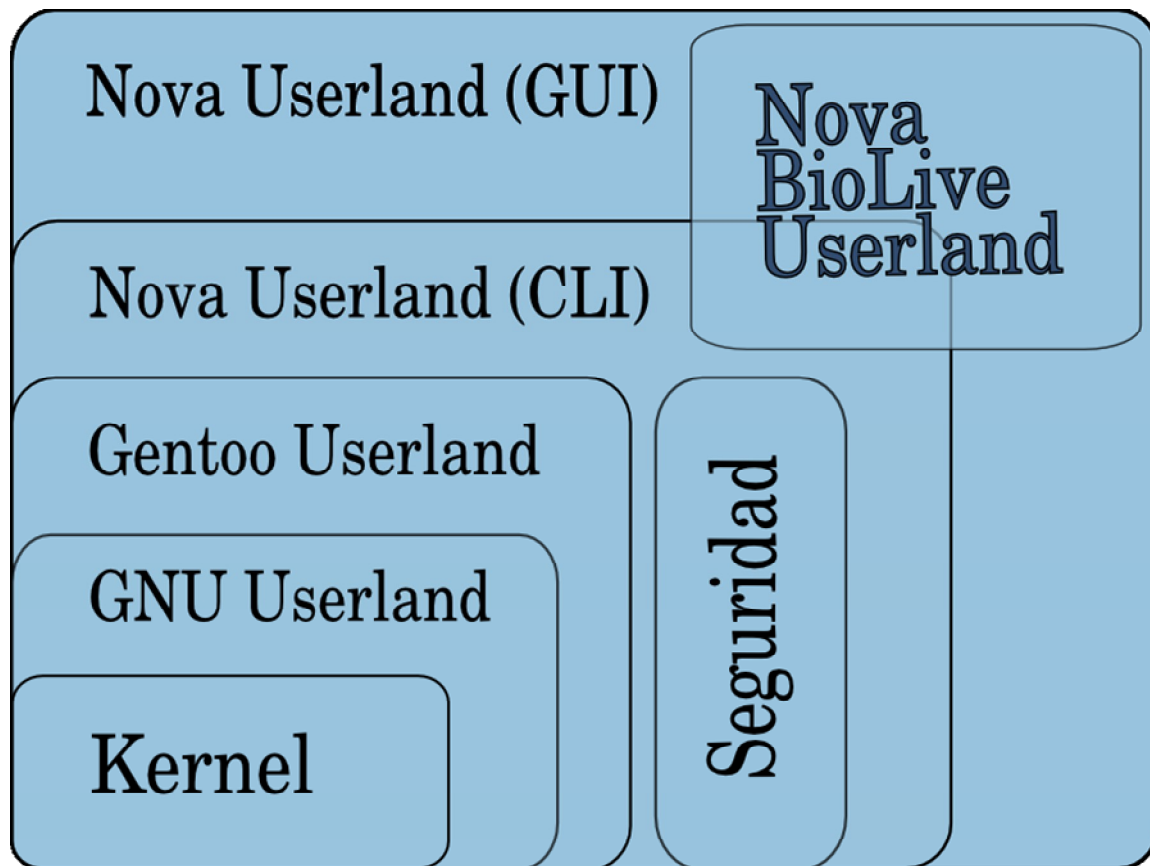


El esquema muestra las transiciones que se realizan en el proceso de arranque de una computadora.

Al encenderse la computadora busca en el primer medio de almacenamiento definido en el BIOS, al encontrar el CD, se lee su primer sector en busca de un gestor de arranque, el cual es ejecutado inmediatamente, encargándose de localizar el kernel en el CD y cargarlo en memoria, este último se

encarga de descomprimir el initrd en la partición virtual creada en la memoria RAM y ejecutar dentro de él un script o aplicación capacitado para detectar la imagen comprimida del sistema a iniciar en el CD, acceder a él, preparar las condiciones para su arranque y por último lo inicia, auto detectando el hardware y autoconfigurando el sistema objeto para que haga un correcto uso de éste.

Anexo 2: Arquitectura de Nova BioLive



Glosario de Términos.

Aplicaciones de open-source: Son aquellas aplicaciones donde el código fuente está disponible, se puede mejorar, distribuir y corregir los errores que presenten.

Bechmarking: Es la tarea de medir la velocidad con la que un ordenador ejecuta procesos, de forma que se puedan realizar comparaciones entre diferentes combinaciones de programas/componentes. Esta definición no tiene en cuenta la sencillez de uso, estética o ergonomía o cualquier otro tipo de juicio subjetivo.

Bioinformática: es la aplicación del desarrollo de la computación y las matemáticas que permite la administración, análisis y comprensión de datos para resolver preguntas biológicas.

Biología de Sistemas: La Biología de Sistemas se centra en estudiar sistemas biológicos en su totalidad, usando herramientas de modelación, simulación, y comparación al experimentado. El foco se tiende hacia sistemas naturales, a menudo con alguna significado medical (por lo menos a lo largo).

Biología Molecular: Es el estudio de la vida a un nivel molecular. Esta área se solapa con otros campos de la Biología y la Química, particularmente Genética y Bioquímica. La Biología Molecular concierne principalmente al entendimiento de las interacciones de los diferentes sistemas de la célula, lo que incluye muchísimas relaciones, entre ellas las del ADN con el ARN, la síntesis de proteínas, el metabolismo, y el cómo todas esas interacciones son reguladas para conseguir un afinado funcionamiento de la célula.

Biotecnología: Toda aplicación tecnológica que utilice sistemas biológicos y organismos vivos o sus derivados para la creación o modificación de productos o procesos para usos específicos.

Cluster de computadoras: Un cluster es un grupo de equipos independientes que ejecutan una serie de aplicaciones de forma conjunta y aparecen ante clientes y aplicaciones como un solo sistema. Los clusters permiten aumentar la escalabilidad, disponibilidad y fiabilidad de múltiples niveles de red.

Diseño de Fármacos: Es la forma de encontrar medicamentos mediante diseños basados en sus acciones biológicas. El objetivo {destino} de un fármaco típicamente es una molécula clave; o una vía

metabólica particular, la cual es específica de una enfermedad o patología; o apunta a la contagiosidad o a la supervivencia de un microbio patógeno.

Distribución de Linux: Una distribución Linux, o distribución GNU/Linux (abreviada con frecuencia distro) es un conjunto de aplicaciones reunidas que permiten brindar mejoras para instalar fácilmente un sistema basado en Linux (también llamado GNU/Linux).

Ebuild: Un ebuild es un procesamiento por lotes especializado creado por el proyecto Gentoo Linux para usarlo con el sistema de mantenimiento de software Portage. Un ebuild es una forma automática de compilar e instalar software.

GNU/Linux: Es la denominación defendida por Richard Stallman y otros para el Sistema operativo que utiliza el kernel Linux en conjunto con las aplicaciones de sistema creadas por el proyecto GNU y de varios otros proyectos/grupos de software. Comúnmente este sistema operativo es denominado como Linux, aunque Stallman sostiene que esta denominación no es correcta.

Infraestructura de soporte: Se orienta a identificar y mejorar el nivel actual de seguridad de los sistemas informáticos y de la red para protegerlos de ataques internos y externos.

LiveCD: sistema operativo (normalmente acompañado de un conjunto de aplicaciones) almacenado en un medio extraíble, tradicionalmente un CD o un DVD, que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro del ordenador.

Metadistribución: distribución hecha a medida por un grupo determinado de usuarios, con un objetivo concreto. Una metadistro es la plataforma para personalizar una distribución GNU/Linux a un entorno de trabajo concreto.

Overhead: es el sobrecoste que toda empresa ha de asumir para que el producto salga a la calle con independencia del salario de los programadores.

Paquetes enmascarados: paquetes que están "testing" o inestables y no se recomienda usarlos. Solo se ponen estables en Gentoo después de 1 mes sin problemas.

Portage: Es el nombre del sistema avanzado de *ports* originario de Gentoo. Es un verdadero sistema de *ports* en la tradición de los ports BSD, pero está basado en Python y posee una serie de características avanzadas, entre ellas: la gestión de dependencias, afinamiento preciso de los paquetes a gusto del administrador, instalaciones falsas al estilo OpenBSD, cajas de arena durante la compilación, desinstalación segura, perfiles de sistema, paquetes virtuales, gestión de los ficheros de configuración y múltiples ranuras para distintas versiones de un mismo paquete.

Química Teórica: Es una herramienta complementaria indispensable para la investigación en el ámbito de la química. Partiendo de la física aplicable a los átomos y moléculas y gracias al espectacular desarrollo de la informática, la Química Teórica puede hoy no solamente interpretar el comportamiento de numerosos sistemas químicos, sino también predecirlos de forma fiable. Por este motivo, su utilización ha rebasado el campo estrictamente académico para convertirse en apoyo de la investigación aplicada en campos industriales punteros, como el de la catálisis o el farmacéutico.

Repositorio: Sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos. Existen repositorios de código fuente y de código binario. Los repositorios están preparados para distribuirse habitualmente sirviéndose de una red informática como Internet, en un medio físico como un CD o a través de un disco duro.

Tecnologías Grid de Computación: Se trata de un modelo de computación que reparte los problemas entre diversos recursos computacionales distribuidos en diversas organizaciones, si no se puede resolver un problema en un único sitio, se reparte entre varios centros de cálculo. La unión de los recursos de varias organizaciones es lo que se llama una grid computacional y actualmente ya se utiliza para mejorar la calidad de muchos proyectos de investigación, como la lucha contra el Cáncer o para la simulación de terremotos.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.