



Facultad de Ingeniería

Escuela de Ingeniería en Bioinformática

BORRADOR Informe

Administración de Sistemas

Alumno:

Kevin Aguilar Valdés

2015430842

kaguilar15@alumnos.utalca.cl

Profesor:

Alejandro Valdés Jimenez

Fecha:

12 de octubre de 2020

Índice

| | |
|---|----------|
| 1. Compilado de programas, librerías, kernel | 5 |
| 1.1. Compilado de programas | 5 |
| 1.2. Librerías | 7 |
| 1.3. Kernel | 7 |
| 1.3.1. Conexión remota a Máquina Virtual | 7 |
| 1.3.2. screen | 7 |
| 1.3.3. Kernel | 7 |
| 1.3.4. Time | 8 |

Índice de figuras

| | |
|---|---|
| 1. Pipeline instalación librerías | 6 |
|---|---|

Índice de cuadros

1. Compilado de programas, librerías, kernel

GNU Compiler Collection (GCC-GNU) es una suite de compiladores desarrollados por el proyecto GNU. GCC es un software libre y permite a los usuarios generar aportes a este, incentivando las mejoras en el compilador. GCC soporta una amplia gama de plataformas, este fomenta el uso de un compilador de alcance global, que atraiga a desarrolladores para asegurar el buen funcionamiento de GCC y de los sistemas GNU en las diferentes arquitecturas y entornos a los cuales puede acceder, para así extender la funcionalidad de esta herramienta.

En esta primera lección se trabajó con la instalación y compilado de programas necesarios para la implementación de GCC, así como la instalación de kernel para el equipo.

1.1. Compilado de programas

En una primera aproximación al compilado de programas se requiere realizar la instalación de una serie de dependencias necesarias para la implementación del compilador GCC. Estas corresponden a las librerías: GMP - MPFR - MPC - ISL.

Para realizar la implementación se utilizó un pipeline de comandos descritos en Figura 1. Se siguió el mismo procedimiento en los cuatro casos, finalizando con la confirmación en la instalación de las librerías. Esto último se logra consultando el directorio `/opt/util_compiler`, en este caso con la utilización del comando `tree`.

```
/opt/util_compiler$ tree lib/  
lib/  
  libgmp.a  
  libgmp.la  
  libisl.a  
  libisl.la  
  libmpc.a  
  libmpc.la  
  libmpfr.a  
  libmpfr.la  
  pkgconfig  
  isl.pc  
  
1 directory, 9 files
```

Una vez concluida la implementación de las librerías se procede a compilar GCC. Este

FORMATO PENDIENTE

- Descarga
 - Se descargan los archivos de las dependencias desde la plataforma web de cada desarrollo.
 - Comando
 - `$ wget.`
- Descompresión
 - Una vez descargados los ficheros se descomprimen
 - Comandos
 - `$ tar`
 - `$ bunzip2`
- Configuración
 - Se ejecuta el script `configure`. Este archivo verifica ciertos aspectos del equipo en que se realizará la instalación. De finalizar satisfactoriamente este paso se crea el archivo `Makefile` en el directorio.
 - Comando `./configure`
 - Parámetro `prefix=/opt/util_compiler/`
 - Este último asigna la ubicación de la instalación de la librería.
- Make
 - Se ejecuta el script que da uso de la información contenida en el archivo `Makefile`. Este contiene la secuencia de acciones a ejecutar para la instalación del programa.
 - Comandos
 - `$ make`
 - `$ make check`
 - Este último permite comprobar la correcta ejecución de `$ make`
- Instalación
 - Finalmente se ejecuta el script que realiza la instalación del programa en el equipo.
 - Comando
 - `$ make install`

Figura 1: Pipeline instalación librerías

Se muestra la serie de pasos seguidos para implementar las dependencias de las librerías en el equipo.

procedimiento sigue la misma lógica expuesta anteriormente. Y se confirma su instalación mediante XXXXX

1.2. Librerías

Para la instalación de librerías se decidió implementar la librería kd-tree.

<http://nuclear.mutantstargoat.com/sw/kdtree/>.

Esta es una dependencia de C trabaja con arboles de decisión de dimensión k. La estructura de datos permite asignar búsquedas rápidas entre nodos estableciendo distancia entre la información que lo compone.

1.3. Kernel

1.3.1. Conexión remota a Máquina Virtual

Para realizar esta actividad se dio acceso a una maquina virtual implementada en un servidor de la Universidad. Para lograr la conexión se utilizó el protocolo ssh. Esta herramienta permite la administración remota de los usuarios para controlar o modificar servidores remotos a través de internet. En este caso se modifico el archivo de configuración ssh con los datos necesarios para acceder a la maquina virtual.

1.3.2. screen

Screen es un multiplexor de terminales, esto significa que es posible generar sesiones en una pantalla y luego iniciar cualquier cantidad de procesos en terminales virtuales, dentro de esta sesión. Los procesos en esta sesión seguirán ejecutándose aun cuando la ventana no sea visible, e incluso cuando se interrumpa la conexión.

El paquete Screen fue implementado en la maquina virtual a través del comando.

```
$ apt install screen
$ screen --version
screen version 4.06.02 (GNU) 23-Oct-17
```

1.3.3. Kernel

Para la implementación de kernel el primer paso es descargar el archivo. Se accede a la pagina de Linux Kernel Archives <https://www.kernel.org>. Se descargo la ultima version: 5.8.14.

```
$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.8.14.tar.xz
$ tar axvf linux-5.8.14.tar.xz
```

X X X X X

Al finalizar el proceso se comprobó que los archivos creados correspondían a los requeridos.

```
root@admSist1:/usr/src# ls -lhtr --color
total 342M
-rw-r--r-- 1 root root 110M Oct  7 03:42 linux-5.8.14.tar.xz
-rw-r--r-- 1 root root 177M Oct 11 11:39 linux-5.8.14_5.8.14.orig.tar.gz
-rw-r--r-- 1 root root 242K Oct 11 11:39 linux-5.8.14_5.8.14-1.diff.gz
-rw-r--r-- 1 root root 1.1K Oct 11 11:40 linux-5.8.14_5.8.14-1.dsc
-rw-r--r-- 1 root root 7.5M Oct 11 14:34 linux-headers-5.8.14_5.8.14-1_amd64
drwxrwxr-x 25 root root 4.0K Oct 11 14:34 linux-5.8.14
-rw-r--r-- 1 root root 1.1M Oct 11 14:34 linux-libc-dev_5.8.14-1_amd64.deb
-rw-r--r-- 1 root root 47M Oct 11 14:36 linux-image-5.8.14_5.8.14-1_amd64.d
-rw-r--r-- 1 root root 5.0K Oct 11 14:36 linux-5.8.14_5.8.14-1_amd64.buildin
-rw-r--r-- 1 root root 2.6K Oct 11 14:36 linux-5.8.14_5.8.14-1_amd64.changes
```

<https://srvbioinf1.utalca.cl/wiki/kernel>

1.3.4. Time

Adicionalmente se utilizó la herramienta en terminal time para determinar el lapso de tiempo en que se generó la compilación.

```
real    178m24.349s
user    167m49.970sGIT
sys     19m48.469s
```