

Garfio

Manual del usuario

Mauro Torres
mauro@garfio.org.ar

GARFIO* v0.1 BETA

MANUAL DEL USUARIO

Autor: Mauro Torres - mauro@garfio.org.ar
Revisión: Gustavo Cortez - gustavo@lugtucuman.org.ar

19 de agosto de 2006

Índice

I	Inicio rápido con GARFIO	2
1.	¿Qué es GARFIO?	3
2.	Características principales	3
3.	Historia	3
4.	Estructura de GARFIO	4
5.	Funcionamiento	4
II	Funciones avanzadas de GARFIO	6
6.	Definiciones previas	6
7.	Funcionamiento general de arranque	6
8.	Funcionamiento de arranque con GARFIO	6
9.	Módulos de GARFIO por defecto	9

Resumen

GARFIO es un sistema de infraestructura muy simple, que fue desarrollado para armar distribuciones a medida, orientada a grupos de usuarios avanzados que requieran un entorno GNU/Linux personalizado, ya sea para universidades, colegios, cursos, proyectos, empresas, etc.

*WebSite: <http://www.garfio.org.ar>

Parte I

Inicio rápido con GARFIO

1. ¿Qué es GARFIO?

GARFIO es un proyecto basado en metadistros, que sirve para armar distribuciones personalizadas muy fácilmente, de una manera estructurada y organizada.

2. Características principales

Algunas características a destacar de GARFIO son

- Su funcionamiento es modulado, lo cual facilita su desarrollo para todo tipo de personas (con algunos conocimientos básicos de Linux), ya que solo deberían agregar o quitar módulos.
- Se encuentra organizado por directorios que lo hace mas visible y expansible.
- Trabaja con variables globales que se pasan como parámetro desde el booteo.
- Dispone de múltiples opciones para la detección de hardware y configuraciones del sistema.
- Arranque detallado y con debugger (depuración) en tiempo real.

GARFIO es actualmente la base de la distribución **Tuquito GNU/Linux**¹.

3. Historia

En mis comienzos con los sistemas de LiveCD y Metadistros, pasé por varias opciones como **Knoopix**, **Morphix** y **Silu**, quedándome con las Metadistros porque tiene algunos mecanismos que consideré bastantes interesantes y claros. Uno de ellos, era el de separar totalmente la distribución con lo que llamarían calzador².

Pero me encontré con varios problemas, ya que no era algo muy estructurado y organizado, entonces decidí hacer uno propio cubriendo algunos de los problemas que estos incluían. De esta manera, arme un conjunto de scripts estructurados llamado GARFIO, que son los ganchos que se van “enganchando” uno a uno.

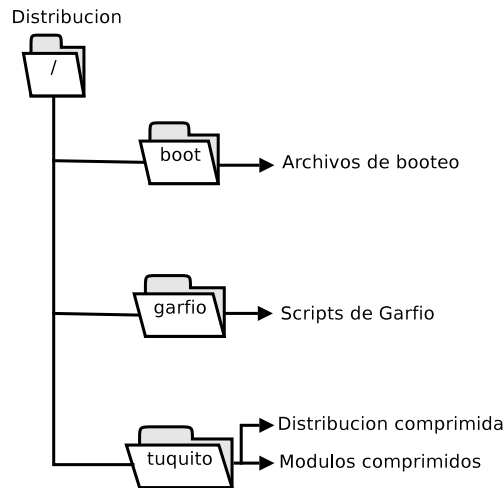
Así como todo proyecto Open Source que surge de la necesidad por resolver problemas específicos, nace GARFIO.

¹Website: <http://www.tuquito.org.ar>

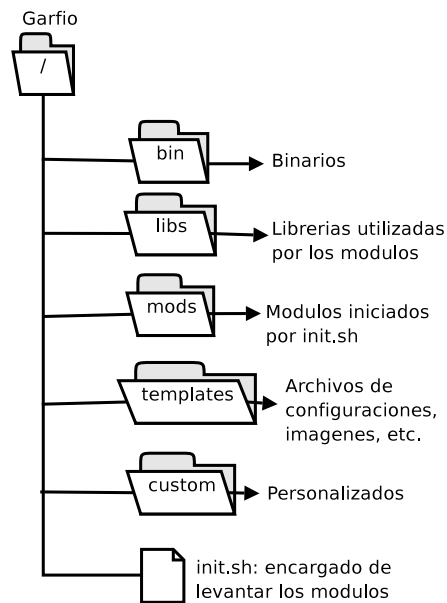
²Calzador es un conjunto de Scripts que tienen como trabajo configurar el *hardware* de la máquina y preparar la distribución realizando la configuración y dejándola lista para el usuario.

4. Estructura de GARFIO

La siguiente figura muestra la estructura principal del CD:



La siguiente figura muestra la estructura de GARFIO:



5. Funcionamiento

¿Cómo puede generar una distribución con GARFIO?

Tiene varios Scripts disponibles en la página de Tuquito³.

- Generador-Auto: es un *frontend* en *bash* muy intuitivo para crear la distro.

³Website: <http://www.tuquito.org.ar>

- **Generador-Manual**: es un archivo donde configura las variables y listo.

Script Generador-Manual

```
### SECCION CONFIGURABLE ###
## Directorios ##
# Directorio donde se encuentra la distribucion linux a comprimir
DIR_SOURCE=/mnt/sources
# Directorio donde se guardara la iso final
DIR_ISO=/mnt/iso
# Directorio donde se encuentra descomprimido Garfio.
DIR_MASTER=/mnt/master
## PATHS ##
# Path del binario mksquashfs
MKSQUASHFS=/usr/bin/mksquashfs
# Path del binario mksquashfs
MKISOFS=/usr/bin/mkisofs
# Path del binario cdrecord
CDRECORD=/usr/bin/cdrecord
## Opciones de grabacion ##
DEV=0,0,0
SPEED=4
## Misc ##
# Nombre de la distro
DISTR0=tuquito
#-----
echo "Comprimiendo la source con squashfs desde $DIR_SOURCE"
$MKSQUASHFS $DIR_SOURCE $DIR_MASTER/TUQUITO/TUQUITO
#-----
echo "Comprimiendo la source con cloop desde $DIR_SOURCE"
#$MKISOFS R L allowmultidot l -
V "${DISTR0} iso9660 filesystem" -
o $DIR_ISO/${DISTR0}_.iso hiderrmved v $DIR_SOURCE
# create_compressed_fs $DIR_ISO/${DISTR0}_.iso 65536 > $DIR_MASTER/TUQUITO/TUQUITO.cloop
# rm rf $DIR_ISO/${DISTR0}_.iso
#-----
echo "Generando la iso final EN $DIR_ISO"
$MKISOFS R b boot/grub/stage2 noemulboot input-
charset ISO88591 V ${DISTR0} \ bootloadsize 4 bootinfo-
table o $DIR_ISO/${DISTR0}.iso $DIR_MASTER
# Blanquear el CD en el caso que sea regrabable
$CDRECORD v $DEV blank=fast
# Grabar el CD
$CDRECORD v $SPEED $DEV $DIR_ISO/${DISTR0}.iso
```

Seguindo el archivo como está configurado, luego descargando GARFIO del sitio oficial, descomprimiéndolo en /mnt/master y colocando mi sistema GNU/Linux a comprimir en el directorio /mnt/sources, seteando las opciones de grabado y listo.

¡A disfrutar de su distribución!

Parte II

Funciones avanzadas de GARFIO

6. Definiciones previas

Initrd Es un archivo especial. El fichero especial */dev/initrd* es un dispositivo de bloque de solo lectura. El dispositivo */dev/initrd* es un disco ram que es inicializado por el gestor de arranque (boot loader).

El núcleo puede usar el contenido de */dev/initrd* para realizar un arranque del sistema en dos fases

En la primera fase el núcleo pone en marcha y monta un sistema de fichero a partir del **initrd**. En la segunda fase carga los módulos adicionales a partir del contenido del *initrd*, ya al finalizar carga el sistema de fichero habitual.

7. Funcionamiento general de arranque

Al encender la máquina, la BIOS se carga y busca el MBR (*Master Boot Record*) en el sector **0** del disco y lanza el gestor de arranque (boot loader).

Cuando se arranca carga el sistema con **initrd** suceden los siguientes pasos

- 1) El gestor de arranque (boot loader) carga el núcleo y el contenido de */dev/initrd* en memoria.
- 2) Al ponerse en marcha el núcleo, este descomprime (si ese es el caso) y copia el contenido del dispositivo */dev/initrd* en el dispositivo */dev/ram0*. Y a continuación libera la memoria ocupada por *initrd*.
- 3) A continuación, el núcleo monta el dispositivo */dev/ram0* como sistema de fichero raíz inicial de modo lectura/escritura.
- 4) Si el archivo ejecutable **linuxrc** se encuentra en dicho sistema, este se ejecuta con **uid 0** (es decir como *superusuario*). El archivo **linuxrc** puede ser cualquier ejecutable válido, incluyendo *shellscripts*. Si el archivo no existe, el núcleo salta al último paso y carga el sistema de archivos habitual.
- 5) Al terminar de leer el archivo **linuxrc**, el núcleo carga el sistema de archivos habitual o el especificado por el **linuxrc**.
- 6) Luego comienza el *arranque/inicio* habitual, por ejemplo con la ejecución de */sbin/init*.

Una vez comprendido el funcionamiento de arranque con *initrd*, pasamos a GARFIO.

8. Funcionamiento de arranque con GARFIO

GARFIO consta de los siguientes pasos para el arranque del sistema



Figura 1: Initrd de GARFIO.

- 1) El gestor de arranque carga el núcleo y el `initrd` en memoria.
- 2) El núcleo descomprime y copia el contenido de `initrd` en `/dev/ram0` y lo monta como sistema raíz. (Ver figura 1).
- 3) Busca el archivo `linuxrc` y lo carga.
- 4) Este archivo se encarga de detectar la unidad de CDROM, encontrar la imagen ISO comprimida y montarla en `/tuquito`. Algunas partes importantes de este archivo son

```

DEVICES="$DEVICES /dev/hd?[19] /dev/hd?[19][09]"
for i in $DEVICES
do
if mountit $i /cdrom "o ro" >/dev/null 2>&1
then
if test d /cdrom/TUQUITO
then
FOUND_META="$i"
break
fi
umount /cdrom
fi
done

```

Busca si hay un CDROM y verifica si corresponde al LiveCD

```

if test n "$FOUND_META" a f /cdrom/TUQUITO/TUQUITO; then
#echo n "Con Squashfs"
mount t squashfs -
o loop /cdrom/TUQUITO/TUQUITO /TUQUITO -
o ro || FOUND_META=""

```

```

elif test n "$FOUND_META" -a -
f /cdrom/TUQUITO/TUQUITO; then
    #echo n "Sin Cloop"
    mount t iso9660 -
o loop /cdrom/TUQUITO/TUQUITO /TUQUITO -
o ro || FOUND_META=""
elif test n "$FOUND_META" -a -
f /cdrom/TUQUITO/TUQUITO.cloop; then
    #echo n "Con Cloop"
    insmod -
f /modules/cloop.o file=/cdrom/TUQUITO/TUQUITO.cloop >/dev/null 2>&1
mountit /dev/cloop /TUQUITO " o ro" || FOUND_META=""
fi

```

Busca la imagen ISO comprimida de la distribución y la monta en /TUQUITO. Esta imagen ISO comprimida contiene toda la distribución instalada.

```

cp -af /TUQUITO/home/usuario /home
# etc
cp -af /etc/* /ramdisk/etc/ && rm -fr /etc && ln -s /ramdisk/etc /
>> /tmp/linuxrc.log 2>&1
# lib
rm -fr /lib && ln -s /TUQUITO/lib / >> /tmp/linuxrc.log 2>&1

```

Copia los archivos importantes al nuevo sistema raíz /dev/ram0, ya que este permite la escritura al igual que los archivos de configuración de /etc.

Los que no necesitan escritura, tales como los binarios de /usr/bin, se los enlaza desde /Tuquito/usr/bin a /usr/bin.

```
ln -s /garfio/init.sh /etc/init.d/rcS 2 >> /tmp/linuxrc.log
```

Indica cual es el archivo que se cargará al finalizar **linuxrc** (/sbin/init) preestablecido en el archivo /etc/inittab.

- 5) Una vez finalizado este Script, se ejecuta **init.sh**, el cual lanza los módulos de GARFIO por orden numérico.

```

## Add vars globals
source /garfio/libs/libglobals.sh
for files in `ls $ROOTDIR/mods/`
do
    $ROOTDIR/mods/$files
done
##

```

Para *tunear* la distribución, como por ejemplo levantar entornos gráficos o cargar servicios, solo se deben agregar los módulos en /mods de GARFIO, que también pueden estar en combinación con librerías que se encuentran en /libs.

9. Módulos de GARFIO por defecto

Lo siguiente es un listado con los módulos utilizados por GARFIO

- 01init.sh** Se encarga de hacer las configuraciones iniciales
- 02dialogs.sh** Revisa si se realizó una llamada a las configuraciones interactivas, y si es así, la carga. Algunas de estas son configuración de red, usuario, contraseña, etc.
- 03hwdetect.sh** Detecta el hardware de la máquina.
- 04partitions.sh** Detecta y monta las particiones ordenándolas por categoría.
- 05xdetect.sh** Detecta y configura la placa de video.
- 06install.sh** Arranca la instalación, si se lo ha indicado desde el arranque.
- 07daemons.sh** Carga los demonios especificados en el archivo.
- 08init.sh** Finaliza las configuraciones y finalmente inicia el entorno gráfico dependiendo de la memoria disponible.