

Parcial 1. Preboot eXecution environment(Pxe) (septiembre de 2020)

Andres Rivera

sndres_miguel.rivera@uao.edu.co

Universidad autónoma de occidente

I. PROCEDIMIENTO

A. Implementación servidor PXE

Para implementar este servidor lo mas importante a tener en cuenta:

- Se debe tener un servidor dhcp configurado para que los computadores que tengan arranque en red se puedan identificar con una dirección ip.
- Se debe contar con un servicio de transporte de archivos, como la transferencia se hace de forma simple solo de archivos se usa un servidor tftp
- Finalmente, para que los usuarios puedan conectarse a un sistema operativo, se debe tener en el servidor pxe algún método para tener montados los archivos para que el computador pueda arrancar. Algunos de estos métodos pueden ser implementar un servicio http(apache), ftp(vsftpd), tftp o un servidor nfs.

Una vez entendidos los tres pasos generales se procede a la instalación del entorno. Respectivamente se procede a instalar los paquetes necesarios.

En Ubuntu, Linux mint y otras distribuciones basadas en Debian se puede instalar los paquetes con apt-get. Para la distribución Centos7 que tenemos que es basada en redhat, los paquetes se instalan así:

Yum install

Finalmente para instalar los paquetes necesarios, como se necesita un servidor dhcp, ftp,tftp,nfs. Además necesitamos instalar syslinux, de este paquete se copiaran unos archivos en específico para poder iniciar cualquier sistema operativo basado en el kernel de GNU Linux..

La instalación de los paquetes necesarios se puede hacer en una línea o línea por línea, en mi caso para hacer mas automatico lo hago en una línea así

```
Sudo yum install vsftpd ftp-server xinetd dhcpd syslinux nfs-
utils -y
```

Una vez instalados todos los paquetes se debe configurar el servicio de dhcp, tftp,vsftpd y nsf.

Configuracion dhcp

Para configurar el dhcp, básicamente se debe configurar el rango de las ips(número de clientes máximos), red con el Gateway que provee el dhcp y la ip del servidor pxe. El archivo de configuración del dhcp se encuentra en /etc/dhcp/dhcpd.conf. Para editarlo se puede abrir con vim o con nano y tener activado el usuario root.

```
sudo vim /etc/dhcp/dhcpd.conf
```

DHCP Server Configuration file.

```
ddns-update-style interim;
ignore client-updates;
authoritative;
allow booting;
allow bootp;
allow unknown-clients;
```

```
# internal subnet for my DHCP Server
subnet 172.168.1.0 netmask 255.255.255.0 {
range 172.168.1.21 172.168.1.151;
option domain-name-servers 172.168.1.11;
option domain-name "pxe.example.com";
option routers 172.168.1.11;
option broadcast-address 172.168.1.255;
default-lease-time 600;
max-lease-time 7200;
```

```
# IP of PXE Server
next-server 172.168.1.11;
filename "pxelinux.0";
}
```

Ahora se procede a configurar el servidor de tftp, para hacerlo se recurre a tener los servicios de xinetd con el cual se administra la conectividad de la red y además de poder configurar listas de acceso (se puede permitir el acceso a la red a solo unos usuarios). Se usara xinetd para tener el servicio de tftp, el archivo de configuración se encuentra en /etc/xinet.d/tftp. Para editarlo se hace de la misma manera con vim o nano. Básicamente lo que se debe editar es que se active como servidor el tftp para hacerlo se configura la línea service tftp

```
{
```

```

socket_type = dgram
protocol    = udp
wait       = yes
user       = root
server     = /usr/sbin/in.tftpd
server_args = -s /var/lib/tftpboot
disable    = no
per_source = 11
cps        = 100 2
flags      = IPv4
}

```

Ahora se procede a copiar los archivos del paquete syslinux, esto es con el objetivo de tener un gestor de arranque del sistema. Los archivos mas importante a copiar son pxelinux(para correr el sistema desde la red), mbooot.c32, menú.c32, chain.c32(para hacer particiones en instalaciones automáticas), vesamenu.c32(que es el que usaremos para visualizar el menú). Para copiar todo desde una línea de código se debe estar en la carpeta /usr/share/syslinux una vez estando ahí se procede a copiar el siguiente comando

```
cp vesamenu.c32 chain.c32 pxelinux.0 memdisk mboot.c32 /var/lib/tftpboot
```

Una vez copiados los archivos de configuración se debe crear la carpeta donde se va a configurar el menú de arranque

```
mkdir /var/lib/tftpboot/pxelinux.cfg
```

Finalmente se debe contar con cualquier imagen iso basada en el kernel de Linux, por ejemplo, en la práctica se usó:

- Clonezilla
- Linux lite
- CentOS7

Entonces se debe montar la imagen y después copiar todos los archivos a la carpeta tftpboot por orden se creo una carpeta denominada data entonces ahí se copia todos los archivos de la imagen CentOS y para la imagen clonezilla se copia todos los archivos en la carpeta var/lib/tftpboot/clonezilla

```

mkdir /var/lib/tftpboot/data
mount -o loop CentOS-7-x86_64-NetInstall-2003.iso /mnt/
cd
cd /mnt

```

```
cp -fr /mnt/* /var/lib/tftpboot/data
```

Una vez copiados todos los archivos se procede a configurar los servicios que van a hacer posible que arranque el sistema operativo para la red. Esto se hace en el archivo var/lib/tftpboot/pxelinux.cfg

Primero configuración tftp

Como ya hemos configurado que el tftp se comporte como servidor se configura la etiqueta de inicio de la siguiente manera

```
MENU LABEL ^2) Clonezilla Live -TFTP
```

```
    KERNEL clonezilla/live/vmlinuz
```

```
    APPEND ramdisk_size=32768
```

```
initrd=clonezilla/live/initrd.img boot=live union=overlay
username=user config components noswap edd=on nomodeset
noefect locales=en_US.UTF-8 keyboard-layouts=NONE
net.ifnames=0 ocs_live_extra_param=""
```

```

ocs_live_keymap="NONE" ocs_live_batch="yes"
ocs_lang="en_US.UTF-8" vga=788 ip=frommedia nosplash
fetch=tftp://172.168.1.11/clonezilla/live/filesystem.squashfs

```

para configurar el servidor vsftpd se debe cambiar el directorio por defecto a /var/lib/tftpboot. Para hacerlo se hace lo siguiente:

```

anonymous_enable=YES
local_enable=NO write_enable=NO
local_umask=022
dirmessage_enable=YES xferlog_std_format=YES
xferlog_enable=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
anon_root=/var/lib/tftpboot/data
anon_max_rate=2048000

```

Para esta configuración se debe tener en cuenta que mínimo se requiere una memoria ram de 2gb.

LABEL 4

```
MENU LABEL ^4) Install CentOS 7 - FTP
```

```
    kernel data/centos7_64Bit/images/pxeboot/vmlinuz
```

```
    append
```

```
initrd=data/centos7_64Bit/images/pxeboot/initrd.img
```

```
inst.repo=ftp://172.168.1.11/centos7_64Bit
```

Finalmente se configura el servidor nfs

Primero se debe crear el directorio donde se compartirá los archivos, carpetas.

```
mkdir /var/nsfshare
```

Después se le da permisos de 755

```
chmod -r 755 /var/nsfshare
```

después se le cambia el dueño y grupo de la carpeta en este caso nsfnobody:nsfnobody

```
chmod nsfnobody:nsfnobody /var/nsfshare
```

Después se inicializan todos los servicios de nsf

```

systemctl enable rpcbind
systemctl enable nfs-server
systemctl enable nfs-lock
systemctl enable nfs-idmap
systemctl start rpcbind
systemctl start nfs-server
systemctl start nfs-lock
systemctl start nfs-idmap

```

finalmente el archivo a donde se comparte los archivos es exports

para hacerlo se puede hacer asi

```
echo "/var/lib/tftpboot/data *(rw)" >> /etc/exports
```

y finalmente se reinicia el servidor nfs

```
systemctl restart nsf.server
```

Finalmente se hace un arranque de red pero teniendo la imagen iso en el cliente. Asi:

LABEL 1

```

MENU LABEL ^1) Boot local hard drive
MENU AUTOBOOT
MENU DEFAULT
LOCALBOOT 0

```

El archive de configuracion de arranque debería quedar de la siguiente manera

```

default vesamenu.c32
prompt 0
timeout 50

```

Local Hard Disk pxelinux.cfg default entry

LABEL 1

```

MENU LABEL ^1) Boot local hard drive
MENU AUTOBOOT
MENU DEFAULT
LOCALBOOT 0

```

LABEL 2

```

MENU LABEL ^2) Clonezilla Live -TFTP
KERNEL clonezilla/live/vmlinuz
APPEND ramdisk_size=32768
initrd=clonezilla/live/initrd.img boot=live union=overlay
username=user config components noswap edd=on nomodeset
noeject locales=en_US.UTF-8 keyboard-layouts=NONE
net.ifnames=0 ocs_live_extra_param=""
ocs_live_keymap="NONE" ocs_live_batch="yes"
ocs_lang="en_US.UTF-8" vga=788 ip=frommedia nosplash
fetch=tftp://172.168.1.11/clonezilla/live/filesystem.squashfs

```

LABEL 3

```

MENU LABEL ^3) Install CentOS 7 - NFS
kernel data/centos7_64Bit/images/pxeboot/vmlinuz
append
initrd=data/centos7_64Bit/images/pxeboot/initrd.img
inst.stage2=nfs:172.168.1.11:/var/lib/tftpboot/data/centos7_64
Bit quiet

```

LABEL 4

```

MENU LABEL ^4) Install CentOS 7 - FTP
kernel data/centos7_64Bit/images/pxeboot/vmlinuz
append
initrd=data/centos7_64Bit/images/pxeboot/initrd.img
inst.repo=ftp://172.168.1.11/centos7_64Bit
Imágenes de la configuración de un cliente en virtual box

```

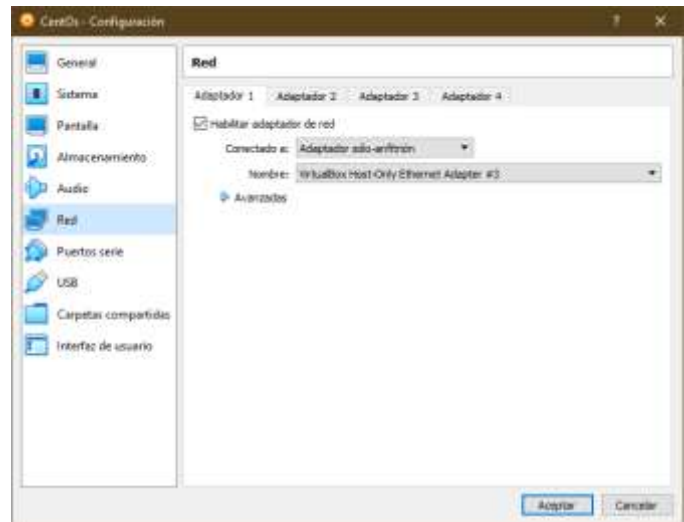


Fig.1 Configuración para estar en la misma red

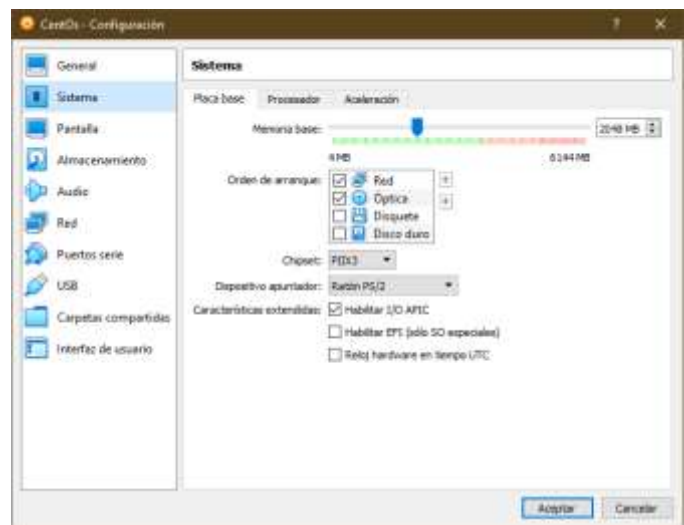


Fig.2 Ram necesaria para correr ftp, los demás con 1024

Algunas imágenes del servidor corriendo

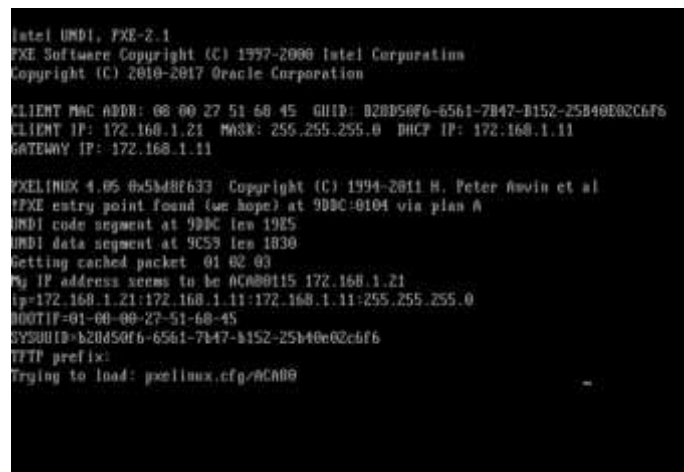


Fig.3 Inicialización dhcp



Fig.4 Menú con col 4 sistemas operativos



Fig. 5 Linux lite con imagen iso en el cliente

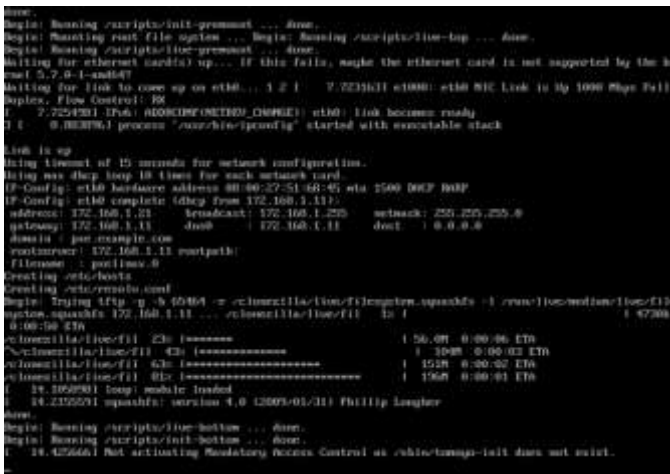


Fig. 6 Clonezilla con tftp



Fig. 7 Interfaz con Clonezilla

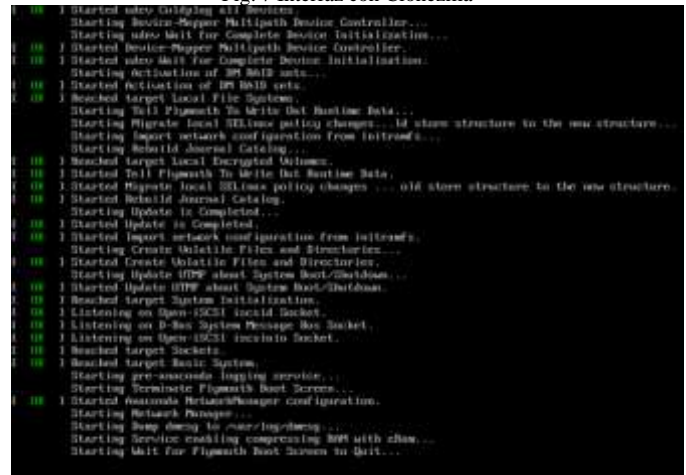


Fig. 8 Centos con nfs



Fig. 9 Centos con nfs

Por ultimo se inicia el la de ftp. La única desventaja es que para correr necesita 2gb de RAM mientras que las anteriores implementaciones solo necesitan 1gb. Tambien se demora mas en iniciar.

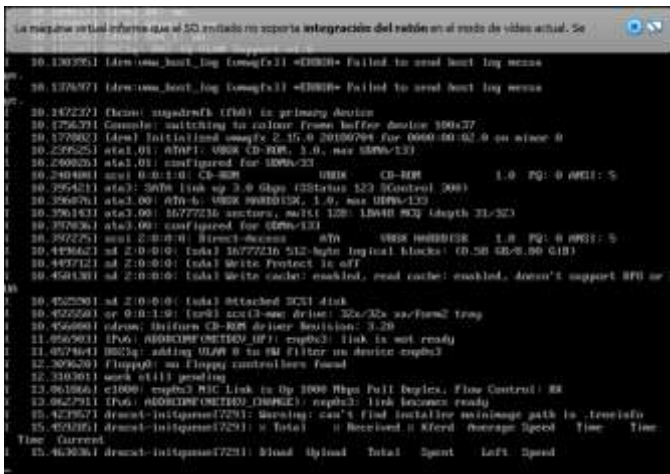


Fig. 10 CentOS con FTP



Fig. 11 Fig.1 Inicialización dhcp
Centos con ftp

B. Punto adicional blockchain

Blockchain es una forma de almacenar información digital

La información almacena en bloques encadenados por hashes
Cadena de bloques

- Inmutabilidad de la historia
- Invulnerabilidad del sistema
- Persistencia de la información

Objetivo de la aplicación

Sitio web que permita Compartir información

Se almacenan los mensajes publicados por cualquier usuario.
Se identifica por el contenido, autor y fecha de publicación en un tipo de dato denominado diccionario en el cual se requiere un atributo y un valor.

Cada bloque tiene un identificador único

Para detectar cualquier fallo se usan los hashes en cada bloque, se usa la función hash. Este hash se usa como si fuera una huella digital de cada bloque. Para evitar fallos de seguridad se generan hashes de forma aleatoria.

El servidor web se implementa usando el framework Flask para crear una REST-API

REFERENCIAS

- [1] L. Topic, «Linux Topic,» 26 Julio 2018. [En línea]. Available: <https://www.linuxtopic.com/2018/07/pxe-boot-server-centos-7.html>. [Último acceso: 15 09 2020].
- [2] HowtoForge, «HowtoForge,» [En línea]. Available: <https://www.howtoforge.com/nfs-server-and-client-on-centos-7>. [Último acceso: 14 09 2020].
- [3] J. Orovengua, «LinuxParty,» [En línea]. Available: <https://www.linuxparty.es/29-internet/10072-configure-el-servidor-de-instalacion-pxe-arranque-de-red-en-centos-7-x>. [Último acceso: 13 09 2020].