

PXE

Instalujemy minimalną wersję Debiana wraz z podstawowymi programami:

```
apt-get install bzip2 unzip p7zip-full mc subversion subversion-tools telnet  
tcpdump ntpdate git rsync net-tools aptitude
```

DHCP

W serwerze DHCP ustawiamy parametry:

```
filename "lpxelinux.0";  
next-server 192.168.0.100;
```

Adres IP należy podmienić na adres naszego serwera.

TFTP

Usługa TFTP jest wykorzystywana przez PXE do botowania przez sieć. Instalujemy:

```
aptitude install tftpd-hpa syslinux pxelinux
```

Kopiujemy i tworzymy:

```
cp /usr/lib/syslinux/memdisk /srv/tftp/  
cp /usr/lib/syslinux/modules/bios/* /srv/tftp/  
cp /usr/lib/PXELINUX/lpxelinux.0 /srv/tftp/  
mkdir /srv/tftp/pxelinux.cfg  
touch /srv/tftp/pxelinux.cfg/default  
touch /srv/tftp/pxelinux.cfg/pxe.conf
```

Edytujemy plik /srv/tftp/pxelinux.cfg/pxe.conf i wpisujemy:

```
MENU TITLE PXE Server  
NOESCAPE 1  
ALLOWOPTIONS 1  
PROMPT 0  
menu width 80  
menu rows 14  
MENU TABMSGROW 24  
MENU MARGIN 10  
menu color border 30;44 #ffffffff #00000000 std
```

Edytujemy plik /srv/tftp/pxelinux.cfg/default i wpisujemy:

```
DEFAULT menu.c32
```

```
TIMEOUT 600
ONTIMEOUT BootLocal
PROMPT 0
MENU INCLUDE pxelinux.cfg/pxe.conf
NOESCAPE 1
LABEL BootLocal
    localboot 0
    TEXT HELP
    Boot to local hard disk
    ENDTEXT
```

Uruchamiamy usługę:

```
/etc/init.d/tftpd-hpa start
```

Botujemy komputer za pomocą sieci. Powinno pojawić się menu po ściągnięciu i wczytaniu konfiguracji przez komputer:



HTTP

Ściągnięcie dużych plików po TFTP jest powolne - protokół HTTP jest szybszy. Instalujemy lekkiego demona HTTPD:

```
aptitude install nginx
```

Kopiujemy plik binarne oraz ISO do katalogu /var/www/html/ - np: ściągniemy sobie z sieci dwa pliki: memtest oraz program do wyświetlania informacji o sprzęcie komputera:

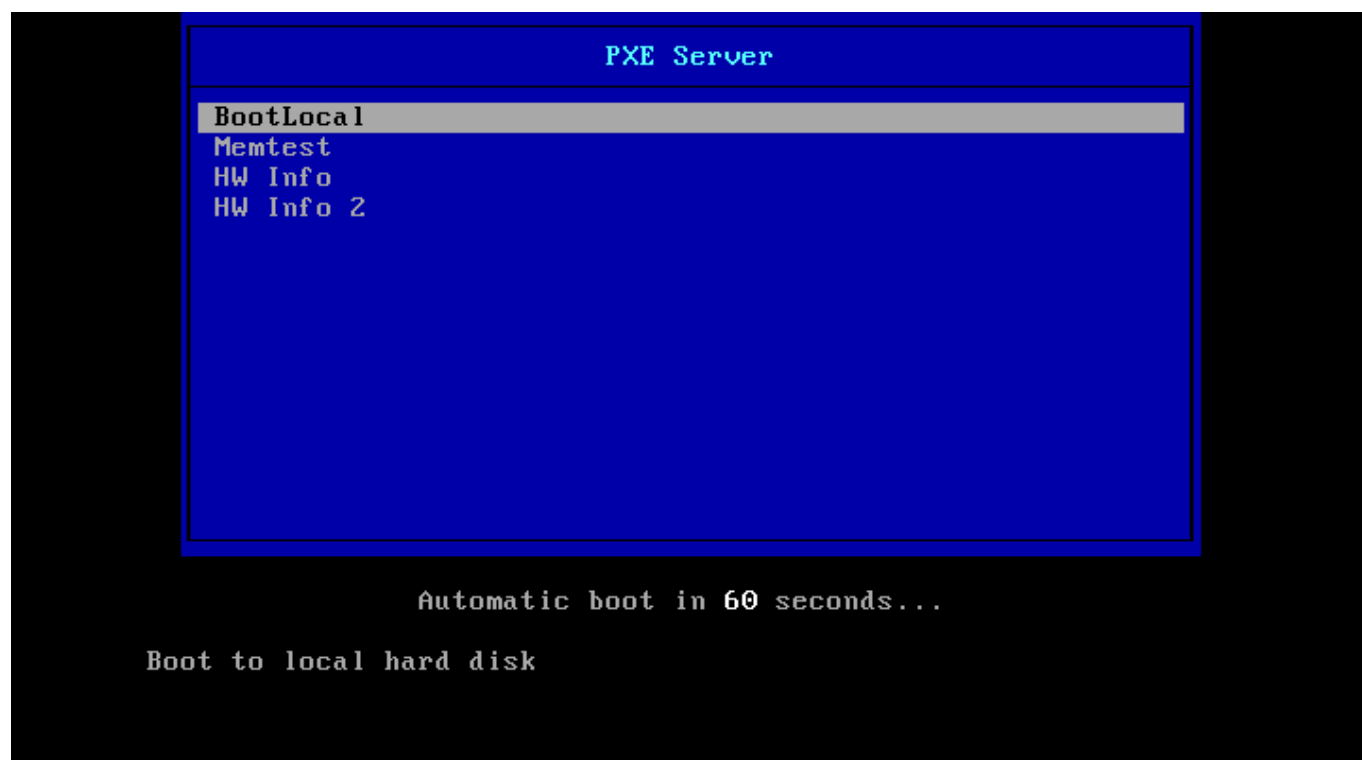
```
wget http://www.memtest.org/download/5.01/memtest86+-5.01.iso.zip
```

```
unzip ./memtest86+-5.01.iso.zip
wget http://www.sysinfo1ab.com/files/astraiso.zip
unzip ./astraiso.zip
wget http://www.serverelements.com/bin/CORE3_Hardware_Detection_Tool.iso
```

Konfigurujemy teraz plik konfiguracji TFTP: /srv/tftp/pxelinux.cfg/default dodając na końcu wpisy:

```
LABEL Memtest
    KERNEL memdisk
    APPEND iso initrd=http://192.168.0.100/memtest86+-5.01.iso raw
    TEXT HELP
    Memory tester
    ENDTEXT
LABEL HW Info
    KERNEL memdisk
    APPEND iso initrd=http://192.168.0.100/astraiso.iso raw
    TEXT HELP
    Hardware info
    ENDTEXT
LABEL HW Info 2
    KERNEL memdisk
    APPEND iso
initrd=http://192.168.0.100/CORE3_Hardware_Detection_Tool.iso raw
    TEXT HELP
    Hardware info 2
    ENDTEXT
```

Adres IP: 192.168.0.100 podmieniamy na adres naszego serwera HTTP.



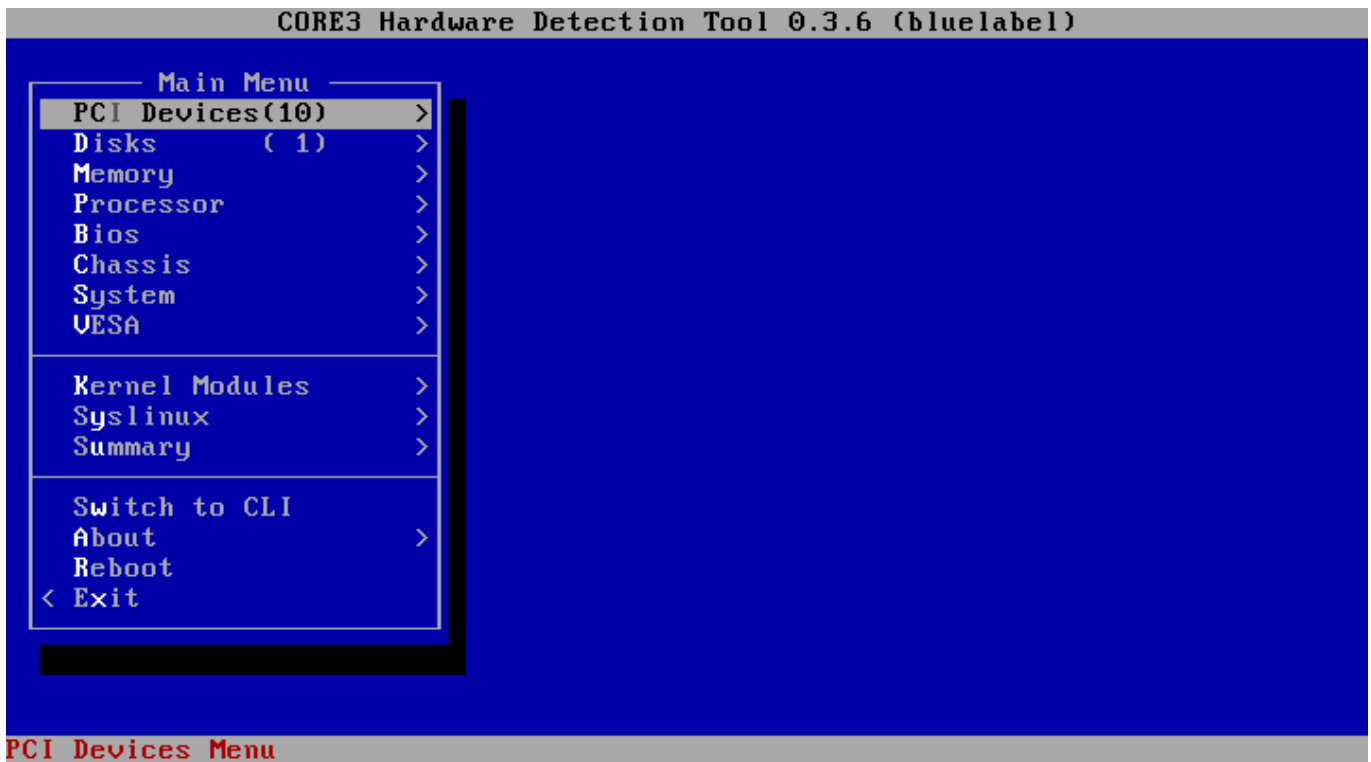
```
Memtest86+ 5.01 : Common KVM processor
CLK: 2400 MHz (32b Mode) : Pass %
L1 Cache: 32K 31580 MB/s : Test %
L2 Cache: 4096K 15190 MB/s : Test #
L3 Cache: None : Testing:
Memory : 2046M : Pattern: : Time: 0:00:00
-----
Core#: : Chipset : Unknown
State: : Memory Type : Unknown
Cores: Active / Total (Run: All) : Pass: 0 Errors: 0
-----

==> Press F1 to enter Fail-Safe Mode <==
==> Press F2 to force Multi-Threading (SMP) <==

(ESC)exit (c)configuration (SP)scroll_lock (CR)scroll_unlock
```

```
Information Reports Help FEATURE-LIMITED DEMO VERSION
----- System Summary TQ0E0* -----
Processor
Processor: Intel Pentium 4 ES
CPU Clock: 2400.1 MHz (1.0 x 2400.1 MHz QDR)
Level 1 Cache: 32 kB + 32k uOp (data+instr.)
Level 2 Cache: 4 MB
Motherboard
BIOS Version: Unknown
Chipset Name: Intel 440FX (Natoma)
Chipset Vendor: Intel Corporation
Chipset: 82441FX 440FX (Natoma) System Controller R
Southbridge Vendor: Intel Corporation
Southbridge: 82371SB PIIX3 ISA Bridge
PCI Bus Version: 2.10
PnP BIOS Version: Not Present
DMI/SMBIOS Version: 2.8
Memory
Base Memory: 640 kB (616 kB free)

F1 Help F2 Report F10 Menu Alt-X Exit | ASTRA Version 6.50 May-22-2017
```



NFS

NFS będzie nam potrzebny do zainstalowania Linuxa Live, np. Debiana lub CentOSa, itp. Instalowanie będzie odbywało się dwójako - najpierw kernel poprzez HTTP, a pozostałe pliki już po NFSie. Instalujemy:

```
aptitude install nfs-kernel-server portmap
```

Konfigurujemy dopisując na końcu pliku `/etc/exports` linijkę:

```
/srv/nfs 192.168.0.0/24(ro,async,no_root_squash,no_subtree_check)
```

Tworzymy katalog oraz przeładowujemy ustawienia:

```
mkdir /srv/nfs
exportfs -a
/etc/init.d/nfs-kernel-server reload
```

Ubuntu

```
wget http://releases.ubuntu.com/18.04.1/ubuntu-18.04.1-desktop-amd64.iso
mkdir /var/www/html/{ubuntu,debian,centos,mint}
mount ./ubuntu-18.04.1-desktop-amd64.iso /mnt
cp /mnt/casper/vmlinuz /srv/install/iso/ubuntu18/
cp /mnt/casper/initrd.lz /srv/install/iso/ubuntu18/
mkdir /srv/nfs/{ubuntu,debian,mint,centos}
cp -r /mnt/* /srv/nfs/ubuntu/
```

SAMBA

Samba będzie nam potrzebna do podmontowania dysku sieciowego, aby móc zainstalować Windowsa przez sieć. Instalacja:

```
aptitude install samba
```

Mój plik /etc/samba/smb.conf wygląda tak:

```
[global]
workgroup = WG
server string = PXE
log file = /var/log/samba/log.%m
max log size = 50
idmap config * : backend = tdb
cups options = raw
netbios name = pxe
map to guest = bad user
dns proxy = no
public = yes
## For multiple installations the same time - not lock kernel
kernel oplocks = no
nt acl support = no
security = user
guest account = nobody

[install]
    path = /srv/samba
    read only = yes
    browseable = yes
    public = yes
    guest ok = yes
```

Tworzymy katalog i restartujemy Sambe:

```
mkdir /srv/samba
/etc/init.d/samba restart
```

Do katalogu /srv/samba kopiujemy katalogi z instalkami Windowsów. Po skopiowaniu nadajemy uprawnienia:

```
chmod -R 777 /srv/samba/*
```

Tworzymy Windows 10 PE wg tutoriala:

<https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-create-usb-bootable-drive> - plik ISO kopiujemy do katalogu /var/www/html/ i dopisujemy konfigurację do pliku /srv/tftp/pxelinux.cfg/default:

```
LABEL Windows 10 PE (64-bit)
    KERNEL memdisk
```

```
INITRD http://192.168.0.100/win10pe_amd64.iso
APPEND iso raw
TEXT HELP
Windows 10 Live CD (64-bit) with installer Windows 7, 8 and 10
ENDTEXT
```

Po załadowaniu Windows 10 PE pojawi nam się konsola, w której wpisujemy:

```
net use z: \\92.168.0.100\install nobody /user:nobody
z:
```

I w ten oto sposób możemy się dostać do instalacji. Wszelkie customizacje Win10PE opisane są w artykule:

<https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-mount-and-customize>

Przykładowy mój plik startnet.cmd do Win10PE:

```
@echo off

wpeinit

:loop
ping 8.8.8.8 -n 1 -4 > nul
if "%errorlevel%" == "1" (
    echo %time:~0,8%: Czekam na adres ip...
    ping 127.0.0.1 -n 3 -4 > nul
    goto loop
)

For /f "tokens=3" %%* in ('route -4 print 0.* ^| find "0.0.0.0"') Do @Set
gw=%%*

if "%gw%" == "1.1.1.1" (
    set ips=1.1.1.16
    set share=install
)
if "%gw%" == "1.1.2.1" (
    set ips=1.1.2.20
    set share=install
)
if "%gw%" == "1.1.3.1" (
    set ips=1.1.3.20
    set share=install
)
if "%gw%" == "1.1.4.30" (
    set ips=1.1.4.18
    set share=pub
)

net use z: \\%ips%\%share% /user:nobody nobody
```

```
Z:
cd Windows

echo Wybierz opcje:
echo 1) Instalator Windows 10 z aktualizacjami
echo 2) Instalator Windows 8 z aktualizacjami
echo 3) Instalator Windows 7 z aktualizacjami
echo 4) Naprawienie bootowania Windowsa
echo Dowolny klawisz) Wyjscie
set /p answer=?
if /i "%answer:~,1%" EQU "1" goto win10
if /i "%answer:~,1%" EQU "2" goto win8
if /i "%answer:~,1%" EQU "3" goto win7
if /i "%answer:~,1%" EQU "4" goto naprawa
goto exit

:win7
    echo Windows 7 z aktualizacjami
    cd 7
    setup.exe
    goto exit

:win8
    echo Windows 8 z aktualizacjami
    cd 8
    setup.exe
    goto exit

:win10
    echo Windows 10 z aktualizacjami
    cd 10
    setup.exe
    goto exit

:naprawa
    echo Nalezy wykonac z poziomu Windows Recovery:
    echo bootrec /fixmbr
    echo bootrec /fixboot
    echo bootrec /rebuildbcd
    goto exit

:exit
```

From:

<https://kamil.orchia.pl/> - **kamil.orchia.pl**

Permanent link:

<https://kamil.orchia.pl/doku.php?id=pxe&rev=1533718382>

Last update: **2018/08/08 10:53**

