

O que é LTS?

O projeto LTS (Servidor de Terminais Linux) está centrado no aproveitamento de micros com pouco poder de processamento para a criação de “terminais burros”, onde a aplicação, um editor de textos por exemplo, está sendo utilizada por um usuário em um micro que não possui HD, CD-Rom e drive de disquete, no qual todo o processamento está sendo feito na verdade pelo servidor devidamente configurado e os dados sendo assim transmitidos para os terminais através da rede.

Este projeto está sendo utilizado pela Prefeitura Municipal de Curitiba, nos pontos de inclusão digital chamados Faróis do Saber. Atualmente estão migrados trinta e quatro Faróis do Saber para o LTS, neste projeto foi considerado que haveria um administrador do sistema, para o gerenciamento de impressão e o desligamento do servidor. Estes servidores estão configurados para que o administrador (usuário “farol” com algumas restrições) e mais oito usuários utilizem o servidor simultaneamente.

Vantagens do LTS.

- Aproveitamento de velhos computadores;
- Pouco custo com hardware;
- Uso de software livre;
- Compartilhamento instantâneo de recursos de rede (impressora, internet, etc);
- Execução de diferentes instancias de programas simultaneamente;
- Armazenamento dos arquivos no servidor, centralizando a informações em uma única unidade de armazenamento;
- Entre outros.

Desvantagens do LTS.

- Performance dos terminais depende extremamente do servidor e da qualidade da rede;
- Placas de vídeos incompatíveis;

Alguns problemas e soluções do LTS.

<i>PROBLEMA ENCONTRADO</i>	<i>SOLUÇÃO PROPOSTA</i>
Congelamento e baixo rendimento de algumas tarefas realizadas pelo terminal	Uma rede de qualidade com velocidade de 100Mb/s (altamente recomendado).
Arquivos alocados na memória sendo corrompidos	Aterramento adequado e uso de No-Break

Material. O material utilizado está baseado no que o Instituto Curitiba de Informática utilizou para implantar o projeto.

Servidor:

- 512MB de memória RAM;
- HD com capacidade de armazenamento de 40GB;
- Processador Celeron 1.8Ghz;
- Duas placas de rede 10/100 sendo uma delas para o compartilhamento de conexão com a internet (placa utilizada RTL8139D);

Terminais:

- 32MB de memória RAM
- Processador Pentium 166Mhz;
- Uma placa de rede 10/100 (placa utilizada RTL8139D);

- EPROM modelo 27C1001;

Outros materiais:

- HUB 16 portas 10/100;
- Cabo UTP 100BaseT;
- Conectores RJ45;

Montagem de terminais.

Como o propósito do projeto é de que os micros iniciem remotamente seria necessário e bastante interessante que fossem retirados: HD, drive de disquete e CD-Rom. Porém a placa mãe desses micros não têm suporte na BIOS para ativar o boot pela rede, então é usado um software que acompanha o disquete do driver da placa de rede RTL8139D, o RSET8139.EXE, ele atualiza a BIOS para este suporte de boot pela rede.

Após ter colocado a placa de rede com a EPROM (supondo que imagem da placa de rede devidamente gravada) é necessário entrar com disquete de boot para fazer a atualização da BIOS, aconselhamos que seja utilizado um disquete de boot do Windows 98 e copiar o arquivo RSET8139.EXE.

Instalação e configuração do servidor.

A partir deste momento apresentaremos como são feitas as configurações do servidor, então, consideramos que já tenha instalado o Debian e os pacotes: vim vim-scripts lynx e ssh, para facilitar a edição de arquivos e a administração remota em seu servidor.

Supondo que tenha alguma máquina com o windows na rede, serão instalados alguns pacotes para o compartilhamento de impressão e internet: SAMBA e SQUID.

Para adiantar a configuração, iremos criar um usuário “administrador” do servidor, que no caso do Instituto Curitiba de Informática será o usuário farol, este será responsável pelo gerenciador de impressão e dependendo da necessidade do usuário, a leitura e/ou cópia de arquivos salvos no servidor. Quanto aos usuários dos terminais serão criados oito usuários, podemos utilizar a seguinte sintaxe:

```
for i in 1 2 3 4 5 6 7 8 ; do adduser --disabled-password micro$i ; done
```

A sintaxe acima executa a rotina para a criação de oito usuários, onde “i” é a variável que está sendo usado pelo laço de repetição, o parâmetro “--disabled-password” adiciona os usuários sem a necessidade de digitar uma senha ao logar-se no sistema, porém, deverá ser feita uma alteração no arquivo “shadow” dentro do diretório /etc, retirando os asteriscos existente entre os “:” (dois pontos), a sintaxe “micro\$i” irá adicionar os usuários “micro” onde “\$i” armazenará a contagem dos usuários, ficando assim:

```
micro1 micro2 micro3 micro4 micro5 micro6 micro7 micro8
```

Nos arquivo group, dentro do diretório /etc é recomendado colocar o colocar o usuário “farol” como “administrador” dos usuários criados.

```
farol:x:1001:  
micro1:x:1002:farol  
micro2:x:1003:farol  
micro3:x:1004:farol  
micro4:x:1005:farol  
micro5:x:1006:farol  
micro6:x:1007:farol  
micro7:x:1008:farol  
micro8:x:1009:farol
```

Para a rede que tenha um micro com windows e seja necessário o compartilhamento de impressora, terá que ser criado um usuário do sistema chamado “smbprint”. Com o seguinte comando:

```
adduser --system --disabled-passowrd smbprint
```

1º passo – Instalação da interface gráfica

A instalação do pacote de gerenciamento gráfico o x-window-system e o qvwm, interface gráfica que será utilizada pelo servidor e terminais na versão stable.

Porque o qvwm? Utilizamos esta interface gráfica, porque ela é extremamente leve nos terminais com 32MB de RAM e não causando também a lentidão da rede, tornando o projeto mais estável e rápido.

```
apt-get install x-window-system qvwm
```

É aconselhável a cópia do arquivo /etc/X11/qvwm/system.qvwmrc para dentro do diretório root, no caso de necessitar de alguma manutenção em perfis dos usuários criados anteriormente.

2º passo - Configuração da área de trabalho dos usuários.

No diretório `/usr/X11R6/lib/X11/qvwm/images` encontram-se as figuras e ícones que, de acordo com a configuração, aparecerá na área de trabalho de cada usuário. Supondo que um dos usuários criados é “administrador” (no nosso caso usuário farol) faremos uma configuração de sua área de trabalho um tanto diferenciada, com acesso aos drives de armazenamento removíveis, gerenciador de impressão e acesso as pastas dos usuários, essa é configuração é feita no arquivo `.qvwmrc`, arquivo oculto o qual armazena informações sobre a configuração da área de trabalho, tais como: ícones das aplicações, atalhos no desktop do usuário, configuração do menu iniciar, etc. Nesse arquivo fica a critério do administrador que está configurando o servidor personalizar. A área de trabalho dos usuários dos terminais ficará, então, no arquivo `/etc/X11/qvwm/system.qvwmrc`, com acesso aos serviços um pouco mais restrito, no nosso caso, os usuários dos terminais ficará apenas acessando o essencial, ou seja, a suíte do OpenOffice, AcrobatReader, Browser, etc.

Para a configuração personalizada do `xdm`, gerenciador da interface gráfica, o arquivo onde encontram-se as suas configurações é o `/etc/X11/xdm/Xresources`, neste arquivo podemos personalizar o login gráfico que aparecerá para os usuários, como por exemplo, o nome do local onde está implantado o projeto.

3º passo – Configuração do SAMBA e SQUID

A instalação dos dois pacotes, o `source.list` deve estar apontando para a versão stable (woody). Essas ferramentas como já foi citado, é responsável pelo compartilhamento dos recursos de rede com máquinas Windows, o SQUID para a conexão com a internet e o SAMBA para o compartilhamento de impressora instalada no servidor.

```
apt-get install samba squid
```

O arquivo de configuração do SAMBA encontra-se no diretório `/etc/samba/`, um arquivo chamado `smb.conf`, a configuração do nosso servidor ficou assim, note que na sessão printer no campo “guest account” está descrito o usuário “smbprint” usuário criado para as impressões de maquinas Windows, segue abaixo como ficou o arquivo de configuração do samba:

```
[global]
    workgroup = farol
    server string = Servidor Farol Guimarães Rosa
    security = SHARE
    encrypt passwords = Yes
    obey pam restrictions = Yes
    passwd program = /usr/bin/passwd %u
    passwd chat = *Enter\snew\sUNIX\spassword:* %n\n
*Retype\snew\sUNIX\spassword:* %n\n .
    syslog = 0
    log file = /var/log/samba/log.%m
    max log size = 1000
    printcap name = cups
    dns proxy = No
    invalid users = root

[homes]
    comment = Home Directories
    create mask = 0700
    directory mask = 0700
    browseable = No

[printers]
    comment = All Printers
    read only = No
    create mask = 0700
    guest ok = Yes
```

```
printable = Yes
guest only = yes
use client driver = yes
guest account = smbprint
path = /home/smbprint
print command = lpr -P %p -o raw %s -r
lpq command = lpstat -o %p
lprm command = cancel %p-%j
printing = cups
```

Na configuração do SQUID, para facilitar a configuração do proxy editamos apenas duas linhas de seu arquivo de configuração, /etc/squid.conf:

Na linha 48 (quarenta e oito) está a especificação da porta liberada para o acesso, então devemos descomentá-la:

```
http_port 3128
```

E finalizando a configuração do proxy, informamos na linha 1444, se o usuário estará liberado para o acesso ou não, que por padrão está comentada e no campo http_access, está o valor “deny”, então descomentaremos a linha e colocar no valor “deny” colocar o valor “allow”:

```
http_access allow all
```

4º passo – Instalação do gerenciador de impressão

Como padrão utilizaremos o Common Unix Printer System (CUPS) da versão stable, então:

```
apt-get install cupsys cupsys-client cupsys-bsd cupsys-driver-gimpprint
```

No término da instalação do CUPS, responderemos “NÃO” para o gerenciamento “BSD”. E é criado também no arquivo /etc/group um grupo chamado “lpadmin”, que por padrão está informado como administrador apenas o usuário “root”, como estamos

considerando que o usuário farol está como “administrador” então após o usuário root devemos acrescentar o usuário farol, ficando assim:

```
lpadmin:x:103:root,farol
```

Obs.: na instalação do CUPS, aconselhamos que seja instalado primeiramente o SAMBA, para evitar um provável conflito, já que os dois pacotes utilizam uma biblioteca chamada “libcupsys2”.

5º passo – Instalando o TFTP e DHCP3

Basicamente na instalação dos pacotes TFTP TFTPd, não há uma configuração alterada, estes pacotes são solicitação para este projeto, pois ele um dos responsáveis pelo boot do terminal, resumindo a instalação dos dois pacotes ficaria assim:

```
apt-get install tftp tftpd
```

Na instalação do pacote DHCP3, existe sim uma configuração personalizada para o arquivo dhcp3.conf, ficando assim:

```
option x-display-manager 192.168.0.254;
option routers 192.168.0.254;
# dhcpd.conf
# Sample configuration file for ISC dhcpd
option domain-name "debian-term.curitiba.pr.gov.br";
option domain-name-servers 192.168.0.254;
#option broadcast-address 192.168.0.255
option subnet-mask 255.255.255.0;
default-lease-time 360000;
max-lease-time 360000;

# bootp options
```

```
allow booting;
allow bootp;
# LTSP Path Options
option root-path "192.168.0.254:/opt/ltsp/i386";
#filename "/tftpboot/lts/bzImage-2.4.22-ltsp-1"; - Comentado para teste de boot
filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";
subnet 192.168.0.0 netmask 255.255.255.0 {
# Dynamic boot workstation
    range dynamic-bootp 192.168.0.30 192.168.0.40;
    use-host-decl-names on;
    range 192.168.0.10 192.168.0.29;
}
subnet 172.16.28.0 netmask 255.255.252.0 {
}
##### Fixa IP to Client #####
group {
use-host-decl-names on;
option log-servers 192.168.0.254;
host ws001 {
hardware ethernet 00:08:54:07:3E:B8;
fixed-address 192.168.0.41;
filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";
}
host ws002 {
hardware ethernet 00:08:54:00:8F:66;
fixed-address 192.168.0.42;
filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";
}
host ws003 {
hardware ethernet 00:08:54:03:9B:13;
fixed-address 192.168.0.43;
filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";
```

```
}  
host ws004 {  
  hardware ethernet 00:08:54:07:84:09;  
  fixed-address 192.168.0.44;  
  filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";  
}  
host ws005 {  
  hardware ethernet 00:08:54:03:9A:F6;  
  fixed-address 192.168.0.45;  
  filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";  
}  
host ws006 {  
  hardware ethernet 00:08:54:06:70:D5;  
  fixed-address 192.168.0.46;  
  filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";  
}  
host ws007 {  
  hardware ethernet 00:08:54:06:8A:BE;  
  fixed-address 192.168.0.47;  
  filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";  
}  
host ws008 {  
  hardware ethernet 00:08:54:07:84:13;  
  fixed-address 192.168.0.48;  
  filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";  
}  
host ws009 {  
  hardware ethernet 00:08:54:00:75:5E;  
  fixed-address 192.168.0.49;  
  filename "/tftpboot/lts/vmlinuz-2.4.22-ltsp-2";  
}  
}
```

6º passo – Instalando o LTSP

Esta é a parte mais importante e mais trabalhosa do projeto, pois os pacotes do LTSP são os responsáveis pela imagem do kernel nos terminais. Começaremos instalando o pacote via dpkg:

```
dpkg -i ltsp-core-i386_3.0.7-3_all.deb
```

No momento da instalação deste pacote, ocorrerá um erro de dependência com os seguintes pacotes: nfs-server, nfs-user-server e nfs-kernel-server, e retornará uma mensagem parecida com essa:

```
dpkg -i ltsp-core-i386_3.0.7-3_all.deb
Selecionando pacote previamente não selecionado ltsp-core-i386.
(Lendo banco de dados ... 27808 arquivos e diretórios atualmente instalados.)
Descompactando ltsp-core-i386 (de ltsp-core-i386_3.0.7-3_all.deb) ...
dpkg: problemas de dependência impedem configuração de ltsp-core-i386:
 ltsp-core-i386 depende de nfs-kernel-server | nfs-user-server | nfs-server; porém:
   Pacote nfs-kernel-server não está instalado.
   Pacote nfs-user-server não está instalado.
   Pacote nfs-server não está instalado.
dpkg: erro processando ltsp-core-i386 (--install):
 problemas de dependência - deixando desconfigurado
Erros foram encontrados durante processamento de:
 ltsp-core-i386
```

Então para resolver este problema de dependência será usado o comando:

```
apt-get -f install
```

Resolvido o problema de dependência, poderemos continuar a instalação dos outros pacotes do LTSP:

```
dpkg -i ltsp-kernel-2.4.22-i386_3.0.12-0_all.deb ltsp-x-core-i386_3.0.4-0_all.deb  
ltsp-x-core-i386_3.0.4-0_all.deb ltsp-x-xserver-fbdev-3.3.6-i386_3.0.0-0_all.deb ltsp-x-  
xserver-svga-3.3.6-i386_3.0.0-0_all.deb
```

Depois da instalação dos pacotes LTSP, é solicitado que execute um script chamado: “ltsp_initialize”. Este script a ser executado é encontrado dentro do diretório /opt/ltsp/templates/.

Feito isso será necessário fazer algumas alterações no arquivo ltsp.conf, deixando o arquivo desta forma:

```
[Default]  
    SERVER          = 192.168.0.254  
    XSERVER          = auto  
    X_MOUSE_PROTOCOL = "PS/2"  
    X_MOUSE_DEVICE   = "/dev/psaux"  
    X_MOUSE_RESOLUTION = 400  
    X_MOUSE_BUTTONS  = 3  
    USE_XFS           = N  
    XkbModel          = abnt2  
    XkbLayout         = br  
    LOCAL_APPS        = N  
    RUNLEVEL          = 5  
    X_MODE_0          = 800x600  
  
#-----  
# Example of specifying X settings for a workstation  
[ws001]  
    XSERVER          = auto  
    LOCAL_APPS        = N  
    USE_NFS_SWAP      = Y  
    SWAPFILE_SIZE     = 32m  
    RUNLEVEL          = 5  
[ws002]  
    XSERVER          = auto
```

LOCAL_APPS = N
USE_NFS_SWAP = Y
SWAPFILE_SIZE = 32m
RUNLEVEL = 5

[ws003]

XSERVER = auto
LOCAL_APPS = N
USE_NFS_SWAP = Y
SWAPFILE_SIZE = 32m
RUNLEVEL = 5

[ws004]

XSERVER = auto
LOCAL_APPS = N
USE_NFS_SWAP = Y
SWAPFILE_SIZE = 32m
RUNLEVEL = 5

[ws005]

XSERVER = auto
LOCAL_APPS = N
USE_NFS_SWAP = Y
SWAPFILE_SIZE = 32m
RUNLEVEL = 5

[ws006]

XSERVER = auto
LOCAL_APPS = N
USE_NFS_SWAP = Y
SWAPFILE_SIZE = 32m
RUNLEVEL = 5

[ws007]

XSERVER = auto
LOCAL_APPS = N
USE_NFS_SWAP = Y

```

        SWAPFILE_SIZE    = 32m
        RUNLEVEL         = 5
[ws008]
        XSERVER          = auto
        LOCAL_APPS       = N
        USE_NFS_SWAP     = Y
        SWAPFILE_SIZE    = 32m
        RUNLEVEL         = 5
[ws009]
        XSERVER          = auto
        LOCAL_APPS       = N
        USE_NFS_SWAP     = Y
        SWAPFILE_SIZE    = 32m
        RUNLEVEL         = 5
# ws004 is my virtual workstation running in a VMware session
#[ws004]
#   DNS_SERVER          = 192.168.0.254
#   XSERVER             = auto
#   X4_BUSID            = "PCI:0:15:0"
#   X_MODE_0            = 800x600
#   LOCAL_APPS          = N
#   USE_NFS_SWAP        = N
#   SWAPFILE_SIZE       = 64m
#   RUNLEVEL            = 5
#-----
# Example of a workstation configured to load some modules
#[ws001]
#   MODULE_01           = agpgart.o           # This is for i810 video
#   MODULE_02           = uart401.o
#   MODULE_03           = sb.o io=0x220 irq=5 dma=1
#   MODULE_04           = opl3.o
#-----

```

```
# Example of ws001 configured for local apps
#[ws001]
# LOCAL_APPS    = Y
# LOCAL_WM      = Y
# NIS_DOMAIN    = ltsp
# NIS_SERVER    = 192.168.0.254

#-----
# Example of a serial printer attached to /dev/ttyS1 on workstation ws001
#[ws001]
# PRINTER_0_DEVICE = /dev/ttyS1
# PRINTER_0_TYPE   = S          # P-Parallel, S-Serial
# PRINTER_0_PORT   = 9100       # tcp/ip port: defaults to 9100
# PRINTER_0_SPEED  = 9600       # baud rate: defaults to 9600
# PRINTER_0_FLOWCTRL = S          # Flow control: S-Software
(XON/XOFF),
#                      # H-Hardware (CTS/RTS)
# PRINTER_0_PARITY  = N          # Parity: N-None, E-Even, O-Odd
#                      # (defaults to 'N')
# PRINTER_0_DATABITS = 8          # Databits: 5,6,7,8 (defaults to 8)
```

7º passo – configurações finais

- Rede

Levando em consideração que estão sendo usadas duas placas de rede, o arquivo `/etc/network/interfaces` deverá ser configurado para duas placas de redes (ETH1 e ETH0) onde uma delas estará configurada na rede dos terminais, ou seja, na rede 192.x.x.x e a outra placa estará na rede da internet (modem, router, etc), ficando o arquivo desta maneira:

```
auto eth0
iface eth0 inet static
```



```
address 192.168.0.254
netmask 255.255.255.0
network 192.169.0.0
broadcast 192.168.0.255

auto eth1
iface eth1 inet static
    address 172.18.4.2
    netmask 255.255.252.0
    network 172.16.28.0
    broadcast 172.18.4.255
    gateway 172.18.4.1
```

- Arquivo Xaccess

O arquivo `/etc/X11/xdm/Xaccess`, deverá ser descomentado na linha 42, ficando assim na linha que será descomentada:

```
*                                #any host can get a login window
```

- Arquivo `inetd.conf`

O arquivo `inetd.conf` guarda algumas informações sobre os serviços inicializados pelo servidor, a linha que será descomentada neste arquivo, é onde estão os parâmetros de boot por FTP, ficando assim:

```
#:BOOT: Tftp service is provided primarily for booting. Most sites
# run this only on machines acting as "boot servers."
tftp      dgram  udp    wait  nobody /usr/sbin/tcpd  /usr/sbin/in.tftpd
/tftpboot
```

- Arquivo `ip_forward`

O arquivo `/proc/sys/net/ipv4/ip_forward` é responsável pelo roteamento quando é usado duas placas de rede, como é o caso do servidor de boot. Este

arquivo possui somente uma linha para ser substituído do valor de “0” para “1”.

- Arquivo sysctl.conf

O arquivo /etc/sysctl.conf é onde é setado algumas variáveis do sistema, nesse arquivo será adicionado a seguinte linha:

```
net/ipv4/ip_forward=1
```

- Arquivo hosts

Arquivo onde são definidas algumas configurações de rede, no nosso caso ficando assim o arquivo:

```
127.0.0.1    localhost.localdomain    localhost
192.168.0.254 farol-29.curitiba.pr.gov.br farol-29
172.18.4.2    farol-29.curitiba.pr.gov.br farol-29
```