SMB HOWTO

作者: David Wood, *dwood@plugged.net.au*
譯者: J.S. Lin, *r6921068@ms.cc.ntu.edu.tw*

v1.0, 10 August 1996

這是 SMB HOWTO {如何使用 SMB}. 這文件描述在 Linux 上如何使用 Session Message Block (SMB) protocol{協定}, 也叫做 NetBIOS 或 LanManager protocol.

目錄

1 介紹

這是 SMB HOWTO {如何使用 SMB} . 這文件描述在 Linux 上如何使用 Session Message Block (SMB) protocol, 也叫做 NetBIOS 或 LanManager protocol.

本文件的維護者是 David Wood (*dwood@plugged.net.au*). 再者,修改或修正可以寄到那,以便下一版包涵進去.

SMB protocol 被使用在 Microsoft Windows 3.11, NT 和 95 用來分享磁碟及印機. 使用 Andrew Tridgell 所做的 Samba 工具套件, UNIX (包含 Linux) 機器能分享磁碟與印表機給 Windows 主機.

吾人使用 Samba 有四樣東西可以做:

1. 分享 linux 磁碟給 Windows 機器.
2. 分享 Windows 磁碟給 linux 機器.
3. 分享 linux 印表機給 Windows 機器.
4. 分享 Windows 印表機給 linux 機器.

這些東西都已包含在本文件中.

Disclaimer: procedures 和 script 要嘛對作者是可以正常運作的,或要嘛由那些(procedures和script)的提供者報告可以正常運作. 以在這裏的資訊做不同的設定可能會不能正常運作.如果你遭遇類似的困境,你可以 e-mail 給作者,提出建議來改進本文件, 但作者並不作任何保證.你預期什麼呢? 作者畢竟只是個顧問 .....

2 進一步資訊

這個 HOWTO 設法去解釋如何設定基本的 SMB檔以及列印服務在 linux 機器上. Samba 是個非常複雜且完整的包裝. 在這個地方沒有要設法去複製所有的 Samba 文件.

進一步資訊,請看以下文件:

- Samba 文件, 可取得於 Samba 發行套件{distribution}的部分. 此發行套件可取得在: *ftp://nimbus.anu.edu.au/pub/tridge/samba/* <ftp://nimbus.anu.edu.au/pub/tridge/samba/>
- The linux Printing HOWTO.
- The Print2Win Mini-HOWTO.

3 安裝

最新 Samba 的原始程式可以取得自:

ftp://nimbus.anu.edu.au/pub/tridge/samba/ <ftp://nimbus.anu.edu.au/pub/tridge/samba/>

然而,如果已安裝 linux 的 Redhat 版本,你可以使用軟體套件{package} 來安裝它.有些其他版本也包含 Samba 的binaries{二元檔程式}.

以下兩個 daemon{管理精靈} 對 Samba 軟體套件 是必要的. 它們一般是被安裝在 /usr/sbin ,而可在開機時,從系統起動的 scripts或inetd來執行. script的例子被展示在 4 (執行 Daemons).

```
smbd (The SMB daemon)
nmbd (提供 NetBIOS nameserver 的支援給 clients)
```

一般而言,以下的 Samba 的 binaries 被安裝在 /usr/bin, 雖然說它的位置是可選擇性.

```
smbclient        (一種供給 UNIX 機器用的 SMB  client{程式})
smbprint         (一種 script 用在列印於 SMB 主機{host}上的列表機)
smbprint.sysv    (同上, 但是提供給 SVR4 UNIX 機器用的)
smbstatus        (列出對當地主機之目前 SMB 連結情況)
smbrun           (一種 'glue' script 用來簡易地執行應用程式於 SMB 主機)
```

再者,一種叫 'print' 的 script 被包函在本 HOWTO 之中, 其用來當作一種有用的 front end{前端介面} 給 smbprint script用.

Samba 軟體套件是簡易安裝的.簡單地從上述的位址取得原始碼, 且閱讀在發行套件中的 README 檔案. 在發行套件中也有一個 docs/INSTALL.txt 檔提供一個簡單的逐步安裝指示.

接著安裝後, 把 daemons 放到 /usr/sbin 而 binaries 放到 /usr/bin. 而 man pages{手冊頁} 擺在 /usr/local/man.

當你在做 Samba 軟體套件, 你在 Makefile 中會有記載組態檔,smb.conf,的位置. 這一般是在 /etc下, 但可以把它擺到你喜歡擺的任意位置. 對於這些使用方法{directions}, 我們將假設你記載組載檔位置成 /etc/smb.conf, log 檔位置爲 log file = /var/log/samba-log.%m ,而 lock 目錄爲 lock directory = /var/lock/samba.

安裝組態{configuration}檔, smb.conf. 去 Samba 所在的的目錄. 看一下在其子目錄 example/simple 而且讀一下 README 檔. 複製在那個目錄下的 smb.conf 檔到 /etc 下. 注意! 如果你有一 linux 發行套件已安裝好 Samba, 你也許已有 Samba 組態檔在 /etc 下. 或許你應該從那個檔開始.

如果你不想把組態檔放在 /etc 目錄下,可以把它擺在你想要擺的地方,然後在 /etc 下做一下 symlink{符號連結}:

```
ln -s /path/to/smb.conf /etc/smb.conf
```

4 執行這兩個 daemon

這兩個 daemon 是 /usr/sbin/smbd 和 /usr/sbin/nmbd.

你可以執行 Samba daemon 從 inetd 或當成單獨的程序. 如果你正設定一個永久檔案伺服器,你應該從 inetd 來執行,所以如果他們死掉,那將重新開始.如果你只是偶爾想要使用 SMB 伺服器,或者用系統管理來輔助,當你需要時,你可以藉由使用 /etc/rc.d/init.d script,或甚至直接手動的.

要從 inetd 來執行 daemon , 請放以下幾行在 inetd 組態檔, /etc/inetd.conf:

```
# SAMBA NetBIOS services (for PC file and print sharing)
netbios-ssn stream tcp nowait root /usr/sbin/smbd smbd
netbios-ns dgram udp wait root /usr/sbin/nmbd nmbd
```

然後下指令以便重新啓動 inetd daemon :

```
kill -HUP 1
```

要從系統啓動的 script 來執行 daemon, 請把以下的 script 置於一個叫做 /etc/rc.d/init.d/smb 檔, 而且 symbolically link 到註釋{comments}所說明的檔案:

```
#!/bin/sh

#
# /etc/rc.d/init.d/smb - starts and stops SMB services.
#
```

```
# The following files should be synbolic links to this file:
# symlinks: /etc/rc.d/rc1.d/K35smb  (Kills SMB services on shutdown)
#           /etc/rc.d/rc3.d/S91smb  (Starts SMB services in multiuser mode)
#           /etc/rc.d/rc6.d/K35smb  (Kills SMB services on reboot)
#

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# See how we were called.
case "$1" in
  start)
    echo -n "Starting SMB services: "
    daemon smbd -D
    daemon nmbd -D
    echo
    touch /var/lock/subsys/smb
    ;;
  stop)
    echo -n "Shutting down SMB services: "
    killproc smbd
    killproc nmbd
    rm -f /var/lock/subsys/smb
    echo ""
    ;;
  *)
    echo "Usage: smb {start|stop}"
    exit 1
esac
```

5 一般組態設定 (/etc/smb.conf)

在 linux (或其他 UNIX機器)上的 Samba 組態設定是被單一檔案, /etc/smb.conf ,所控制. 這個檔決定你想要哪個系統資源分享給外面世界以及你想對他們做什麼限制.

因爲以下章節將會處理分享 linux 磁碟與印表機給 Windows 機器, 本小節所展示的 smb.conf 檔儘量是你所能取得較簡單的, 它只是以介紹性爲目的.

目前還不要擔心細節部分. 後面的章節將會介紹主要的概念.

在檔案中的每的段落{section}會有個段落標頭開始, 其標頭像是 [global], [homes], [printers] 等等.

[global] 段落定義一些變數, 而 Samba 將用來定義於所有資源之分享.

[homes] 段落允許遠端使用者來存取在當地(linux)機器上之他們的(且只有他們的) home directories{家目錄}. 也就是, 如果一個 Windows 使用者從他們的 Windows 機器要設法連接到這個分享, 他們將是連接

到他們自己個人的 home directiory. 要做這樣事情,所要注意的是,他們必須在linux box上有個帳號.

下的 smb.conf 檔案樣本允許遠端使用者來用當地機器上他們的 home directories 而且寫到一個暫存的目錄. 對一個 Windows 使用者,要想看到這些分享, linux box 必須是在區域網路{local network}內. 然後,使用者簡單地從 Windows File Manager 或 Windows Explorer 連接到網路磁碟.

在以下段落要注意的是, 對這個檔之額外項目將會用來允許更多資源被分享.

```
; /etc/smb.conf
;
; Make sure and restart the server after making changes to this file, ex:
; /etc/rc.d/init.d/smb stop
; /etc/rc.d/init.d/smb start

[global]
; Uncomment this if you want a guest account
; guest account = nobody
   log file = /var/log/samba-log.%m
   lock directory = /var/lock/samba
   share modes = yes

[homes]
   comment = Home Directories
   browseable = no
   read only = no
   create mode = 0750

[tmp]
   comment = Temporary file space
   path = /tmp
   read only = no
   public = yes
```

6 分享 Linux 磁碟給 Windows 機器

以上所展示的 smb.conf 樣本檔, 分享 linux 磁碟給 Windows 使用者是很容易的. 然而, 像 Samba 其他的東西, 你可以控制到很大的程度. 以下有些例子 :

為了分享目錄給大家使用, 就建立個類似上述 [tmp] 段落, 亦即加入像這個的東西到 smb.conf:

```
[public]
   comment = Public Stuff
   path = /home/public
   public = yes
   writable = yes
   printable = yes
```

為了使以上目錄供大眾可讀取,但只有群組成員{group staff}可寫入,請修改其項目像這樣:

```
[public]
   comment = Public Stuff
   path = /home/public
```

```
    public = yes
    writable = yes
    printable = no
    write list = @staff
```

其他玩磁碟分享的技巧,請看 Samba 文件或 man pages.

7 分享 Windows 磁碟給 Linux 機器

一種給 UNIX 機器用的 SMB client 程式被包含在 Samba 發行套件之中.它提供一種在命令列下的像ftp的介面.

想要看在給定的主機上有哪些分享可取得, 執行:

```
/usr/sbin/smbclient -L host
```

這裏的 'host' 是你想要瀏覽的機器名子. 這將會回應一串 'service' 的名稱 - 也就是, 它所分享給你的磁碟或印表機的名稱. 除非 SMB 伺服器沒有安全上的設定, 否則它將要求你輸入密碼. 給它 'guest' 帳號的密碼或是你個人在那台機器上帳號的密碼.

舉個例子:

```
smbclient -L zimmerman
```

這個指令的輸出應該看起來像是這樣:

```
 Server time is Sat Aug 10 15:58:27 1996
 Timezone is UTC+10.0
 Password:
 Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]

 Server=[ZIMMERMAN] User=[] Workgroup=[WORKGROUP] Domain=[]

         Sharename      Type      Comment
         ---------      ----      -------
         ADMIN$         Disk      Remote Admin
         public         Disk      Public
         C$             Disk      Default share
         IPC$           IPC       Remote IPC
         OReilly        Printer   OReilly
         print$         Disk      Printer Drivers


 This machine has a browse list:

         Server               Comment
         ---------            -------
         HOPPER               Samba 1.9.15p8
         KERNIGAN             Samba 1.9.15p8
```

```
        LOVELACE             Samba 1.9.15p8
        RITCHIE              Samba 1.9.15p8
        ZIMMERMAN
```

這個 browse list 列出其他有資源分享給網路的 SMB 伺服器.

使用此 client, 請執行:

```
    /usr/sbin/smbclient service <password>
```

這裏的 'service' 是一台機器和分享名稱. 舉例而言,如果你設法連到一台叫做 zimmerman 的機器上之以 'public' 爲分享名稱的目錄 ,此 service 要叫做 \\zimmerman\public. 然而,因爲 shell 限制, 你會需要脱離 backslashes, 所以你要像這樣的結束:

```
    /usr/sbin/smbclient \\\\zimmerman\\public mypasswd
```

這樣的 'mypasswd' 是你的密碼之文字串{literal string}.

你將會有 smbclient 的提示符號{prompt}出現:

```
     Server time is Sat Aug 10 15:58:44 1996
     Timezone is UTC+10.0
     Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]
     smb: \>
```

鍵入 'h' 取得關於使用 smbclient 的 help:

```
     smb: \> h
     ls             dir            lcd            cd             pwd
     get            mget           put            mput           rename
     more           mask           del            rm             mkdir
     md             rmdir          rd             prompt         recurse
     translate      lowercase      print          printmode      queue
     cancel         stat           quit           q              exit
     newer          archive        tar            blocksize      tarmode
     setmode        help           ?              !
     smb: \>
```

如果你能使用ftp,你應該不需要查詢 smbclient 的 man pages.

8 分享 Linux 印表機給 Windows 機器

爲了分享 Linux 印表機給 Windows 機器, 你必須確定你的印表機已經裝設好. 如果你能從 linux 上列印 ,設定 SMB 的印表機分享將會很直接.

請見 Printing HOWTO 來設定當地的列印.

因爲作者使用被連接到 Windows NT 機器的印表機, 這小節不應該被視爲很完整的,而只是建議.任何有詳細資料可分享出來的人, 請將這些資料寄到 *dwood@plugged.net.au* 使得這小節能夠完成.

加入列印設定到你的 smb.conf:

```
[global]
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes
   log file = /var/log/samba-log.%m
   lock directory = /var/lock/samba

[printers]
   comment = All Printers
   security = server
   path = /var/spool/lpd/lp
   browseable = no
   printable = yes
   public = yes
   writable = no
   create mode = 0700

[ljet]
   security = server
   path = /var/spool/lpd/lp
   printer name = lp
   writable = yes
   public = yes
   printable = yes
   print command = lpr -r -h -P %p %s
```

確認印表機的路徑(本例子中是在[ljet]下)要與 /etc/printcap中的 spool 目錄相符合!

注意: 使用 Samba 來分享 UNIX boxes 上的印表機給 Windows NT 機器有一些問題. 其中一個問題是使用 NT 適當地看到分享的印表機. 想要修正這個問題, 請看 Samba 發行套件上的說明, 即 docs/WinNT.txt 檔案.其他是處理密碼的問題. 對於這擾人問題的了解與不能成功的修正,請看相同文件上的說明.

9 分享 Windows 印表機給 Linux 機器

爲了分享 Windows 機器上的印表機, 你必須做以下步驟:

a) 你必須在 /etc/printcap 內有適當的記錄,而且他們必須與當地目錄結構 (像 spool 目錄等) 相符合.

b) 你必須有 script 檔 /usr/bin/smbprint. 這會隨附在 Samba source, 但不是隨附於所有 Samba binary 發行套件. 一個稍微修改版本在以下討論.

c) 如果你想要轉換 ASCII 檔到 Postscript,你必須有netscript,或與其等效的. netscipt 是一種 Postscript 轉換器而且一般是安裝在 /usr/bin.

d) 你可以希望藉由有易於使用的 frond end 來使 Samba 列印更容易. 一種用於處理 ASCII, Postscript 或建立 Postscript 的簡單 perl script 在下面給定.

以下 /etc/printcap 項目用在 Windows NT 主機上 HP 5MP 印表機. 這些項目如下:

```
    cm - comment
         {說明}
```

```
lp - device name to open for output
     {為了輸出,而要開啓的設備名稱}
sd - the printer's spool directory (on the local machine)
     {印表機的 spool 目錄 (在當地的機器上)}
af - the accounting file
     {帳號檔案}
mx - the maximum file size (zero is unlimited)
     {最大檔案大小 (零表示沒限制)}
if - name of the input filter (script)
     {輸入過濾之名稱 (script)}
```

要取得更多資訊,請看 Printing HOWTO 或者關於 printcap 的 man page.

```
# /etc/printcap
#
# //zimmerman/oreilly via smbprint
#
lp:\
        :cm=HP 5MP Postscript OReilly on zimmerman:\
        :lp=/dev/lp1:\
        :sd=/var/spool/lpd/lp:\
        :af=/var/spool/lpd/lp/acct:\
        :mx#0:\
        :if=/usr/bin/smbprint:
```

請確定 spool 和 accounting{帳號} 的目錄存在且可寫入. 請確信某行有適當的路徑指到 smbprint script (在下面會給) 而且確定適當的設備被指到 (/dev 下某個檔).

下個是 smbprint script 本身. 它通常被擺在 /usr/bin 而且是因為 Andrew Tridgell 的緣故,而據我所知是他創造了 Samba. 它隨附在 Samba source 發行套件, 但在某些 binary 發行套件並沒有, 所以在這我再造一個.

你可以希望仔細地看這個. 有某些較小的修改已經證明它們是很有用的.

```
#!/bin/sh -x

# This script is an input filter for printcap printing on a unix machine. It
# uses the smbclient program to print the file to the specified smb-based
# server and service.
# For example you could have a printcap entry like this
#
# smb:lp=/dev/null:sd=/usr/spool/smb:sh:if=/usr/local/samba/smbprint
#
# which would create a unix printer called "smb" that will print via this
# script. You will need to create the spool directory /usr/spool/smb with
# appropriate permissions and ownerships for your system.

# Set these to the server and service you wish to print to
# In this example I have a WfWg PC called "lapland" that has a printer
# exported called "printer" with no password.
```

```
#
# Script further altered by hamiltom@ecnz.co.nz (Michael Hamilton)
# so that the server, service, and password can be read from
# a /usr/var/spool/lpd/PRINTNAME/.config file.
#
# In order for this to work the /etc/printcap entry must include an
# accounting file (af=...):
#
#   cdcolour:\
#       :cm=CD IBM Colorjet on 6th:\
#       :sd=/var/spool/lpd/cdcolour:\
#       :af=/var/spool/lpd/cdcolour/acct:\
#       :if=/usr/local/etc/smbprint:\
#       :mx=0:\
#       :lp=/dev/null:
#
# The /usr/var/spool/lpd/PRINTNAME/.config file should contain:
#   server=PC_SERVER
#   service=PR_SHARENAME
#   password="password"
#
# E.g.
#   server=PAULS_PC
#   service=CJET_371
#   password=""

#
# Debugging log file, change to /dev/null if you like.
#
logfile=/tmp/smb-print.log
# logfile=/dev/null


#
# The last parameter to the filter is the accounting file name.
#
spool_dir=/var/spool/lpd/lp
config_file=$spool_dir/.config

# Should read the following variables set in the config file:
#   server
#   service
#   password
#   user
eval `cat $config_file`

#
# Some debugging help, change the >> to > if you want to same space.
#
echo "server $server, service $service" >> $logfile
```

```
(
# NOTE You may wish to add the line `echo translate' if you want automatic
# CR/LF translation when printing.
        echo translate
        echo "print -"
        cat
) | /usr/bin/smbclient "\\\\$server\\$service" $password -U $user -N -P >> $logfile
```

大部分 linux 發行套件隨附 nenscript 其用來轉換 ASCII 文件成 Postscript. 以下 perl script 經由 smbprint 使更容易提供一種簡單介面給 linux 的列印.

```
Usage: print [-a|c|p] <filename>
       -a prints <filename> as ASCII
       -c prints <filename> formatted as source code
       -p prints <filename> as Postscript
        If no switch is given, print attempts to
        guess the file type and print appropriately.
```

使用 smbprint 來列印 ASCII 檔案便於截斷過長的行. 如果可能的話,這個 script 截斷長行於空白鍵處(取代在字的中間).

格式化過的原始碼是被 nenscript 處理過. 它處理 ASCII 檔而且用個特選的標頭(像 date,filename等)格式化成兩欄. 它也可計算行數. 使用這當作例子, 其他格式型態可以達成的.

Postscript 文件已經適當地格式化了,所以它們直接通過.

```
#!/usr/bin/perl

# Script:   print
# Authors:  Brad Marshall, David Wood
#           Plugged In Communications
# Date:     960808
#
# Script to print to oreilly which is currently on zimmerman
# Purpose:  Takes files of various types as arguments and
# processes them appropriately for piping to a Samba print script.
#
# Currently supported file types:
#
# ASCII      - ensures that lines longer than $line_length characters wrap on
#              whitespace.
# Postscript - Takes no action.
# Code       - Formats in Postscript (using nenscript) to display
#              properly (landscape, font, etc).
#

# Set the maximum allowable length for each line of ASCII text.
$line_length = 76;

# Set the path and name of the Samba print script
$print_prog = "/usr/bin/smbprint";
```

```
# Set the path and name to nenscript (the ASCII-->Postscript converter)
$nenscript = "/usr/bin/nenscript";

unless ( -f $print_prog ) {
        die "Can't find $print_prog!";
}
unless ( -f $nenscript ) {
        die "Can't find $nenscript!";
}

&ParseCmdLine(@ARGV);

# DBG
print "filetype is $filetype\n";

if ($filetype eq "ASCII") {
        &wrap($line_length);
} elsif ($filetype eq "code") {
        &codeformat;
} elsif ($filetype eq "ps") {
        &createarray;
} else {
        print "Sorry..no known file type.\n";
        exit 0;
}
# Pipe the array to smbprint
open(PRINTER, "|$print_prog") || die "Can't open $print_prog: $!\n";
foreach $line (@newlines) {
        print PRINTER $line;
}
# Send an extra linefeed in case a file has an incomplete last line.
print PRINTER "\n";
close(PRINTER);
print "Completed\n";
exit 0;

# ---------------------------------------------------- #
#        Everything below here is a subroutine         #
# ---------------------------------------------------- #

sub ParseCmdLine {
        # Parses the command line, finding out what file type the file is

        # Gets $arg and $file to be the arguments (if the exists)
        # and the filename
        if ($#_ < 0) {
                &usage;
        }
        # DBG
#       foreach $element (@_) {
```

```
#               print "*$element* \n";
#       }

        $arg = shift(@_);
        if ($arg =~ /\-./) {
                $cmd = $arg;
        # DBG
#       print "\$cmd found.\n";

                $file = shift(@_);
        } else {
                $file = $arg;
        }

        # Defining the file type
        unless ($cmd) {
                # We have no arguments

                if ($file =~ /\.ps$/) {
                        $filetype = "ps";
                } elsif ($file =~ /\.java$|\.c$|\.h$|\.pl$|\.sh$|\.csh$|\.m4$|\.inc$|\.html$|\.htm$/) {
                        $filetype = "code";
                } else {
                        $filetype = "ASCII";
                }

                # Process $file for what type is it and return $filetype
        } else {
                # We have what type it is in $arg
                if ($cmd =~ /^-p$/) {
                        $filetype = "ps";
                } elsif ($cmd =~ /^-c$/) {
                        $filetype = "code";
                } elsif ($cmd =~ /^-a$/) {
                        $filetype = "ASCII"
                }
        }
}

sub usage {
        print "
Usage: print [-a|c|p] <filename>
       -a prints <filename> as ASCII
       -c prints <filename> formatted as source code
       -p prints <filename> as Postscript
        If no switch is given, print attempts to
        guess the file type and print appropriately.\n
";
        exit(0);
}
```

```
sub wrap {
        # Create an array of file lines, where each line is < the
        # number of characters specified, and wrapped only on whitespace

        # Get the number of characters to limit the line to.
        $limit = pop(@_);

        # DBG
        #print "Entering subroutine wrap\n";
        #print "The line length limit is $limit\n";

        # Read in the file, parse and put into an array.
        open(FILE, "<$file") || die "Can't open $file: $!\n";
        while(<FILE>) {
                $line = $_;

                # DBG
                #print "The line is:\n$line\n";

                # Wrap the line if it is over the limit.
                while ( length($line) > $limit ) {

                        # DBG
                        #print "Wrapping...";

                        # Get the first $limit +1 characters.
                        $part = substr($line,0,$limit +1);

                        # DBG
                        #print "The partial line is:\n$part\n";

                        # Check to see if the last character is a space.
                        $last_char = substr($part,-1, 1);
                        if ( " " eq $last_char ) {
                            # If it is, print the rest.

                            # DBG
                            #print "The last character was a space\n";

                            substr($line,0,$limit + 1) = "";
                            substr($part,-1,1) = "";
                            push(@newlines,"$part\n");
                        } else {
                             # If it is not, find the last space in the
                             # sub-line and print up to there.

                            # DBG
                            #print "The last character was not a space\n";

                             # Remove the character past $limit
                             substr($part,-1,1) = "";
```

```
                                # Reverse the line to make it easy to find
                                # the last space.
                                $revpart = reverse($part);
                                $index = index($revpart," ");
                                if ( $index > 0 ) {
                                  substr($line,0,$limit-$index) = "";
                                  push(@newlines,substr($part,0,$limit-$index)
                                      . "\n");
                                } else {
                                  # There was no space in the line, so
                                  # print it up to $limit.
                                  substr($line,0,$limit) = "";
                                  push(@newlines,substr($part,0,$limit)
                                      . "\n");
                                }
                        }
                }
                push(@newlines,$line);
        }
        close(FILE);
}

sub codeformat {
        # Call subroutine wrap then filter through nenscript
        &wrap($line_length);

        # Pipe the results through nenscript to create a Postscript
        # file that adheres to some decent format for printing
        # source code (landscape, Courier font, line numbers).
        # Print this to a temporary file first.
        $tmpfile = "/tmp/nenscript$$";
        open(FILE, "|$nenscript -2G -i$file -N -p$tmpfile -r") ||
                die "Can't open nenscript: $!\n";
        foreach $line (@newlines) {
                print FILE $line;
        }
        close(FILE);

        # Read the temporary file back into an array so it can be
        # passed to the Samba print script.
        @newlines = ("");
        open(FILE, "<$tmpfile") || die "Can't open $file: $!\n";
        while(<FILE>) {
                push(@newlines,$_);
        }
        close(FILE);
        system("rm $tmpfile");
}

sub createarray {
        # Create the array for postscript
```

```
            open(FILE, "<$file") || die "Can't open $file: $!\n";
            while(<FILE>) {
                    push(@newlines,$_);
            }
            close(FILE);
    }
```

10 版權

11 感激

只要你寫信給我提供意見, 我將會在這(文件)的下一版感激你.