NetWare Loadable Module Programming HOWTO

Table of Contents

Table of Contents

NetWare Loadable Module Programming HOWTO

Martin Hinner, < mhi@penguin.cz>

Version 0.2, 04 August 2000

This document describes how to develop NetWare Loadable Modules under Linux, using GNU CC and nlmconv(1) from GNU binutils. This is not an official Novell document; I wrote it without any help or cooperation from Novell, Inc.

4.NLM Header file

5.Message files

6.Help Files

7.XDC Data Files

8.Header Files (.h)

9.Import Files (.imp)

1.Introduction

NetWare Loadable Modules (NLMs) are programs which run on Novell NetWare server. NLMs become part of the NetWare OS. You can load and unload NLMs while the server is running.

"Official" compilers for NLMs are:

- Watcom C/C++
- Metrowerks Codewarrior for NetWare (see http://www.metrowerks.com/)
- EPC C/C++ (see http://www.epc.com)
- Novell NLMLINK.EXE

(On a side note, NetWare 5 can also load 32bit DLLs, which can be built using Microsoft Visual C++, Borland C++ and other Windows compilers. For more information see http://developer.novell.com/ndk/dllcomp.htm)

This document describes how to get started with NLM development under Linux (and possibly other Unixes). Please note that this project is in very early stages of development, so a lot of things may not work as you'd expect.

This document assumes that you are familiar with Novell NetWare, and that you have at least basic knowledge of writing NLMs. For more information about writing NLMs, see Novell's developer site, http://developer.novell.com/. You should also have experience with Unix and C/C++ programming with GNU CC. You can find a lot of information about this topic at http://www.linuxdoc.org/.

1.1 C++ Development

As far as I know, C++ development with gcc is currently impossible, till somebody ports at least the libstdc++ and libgcc libraries from the gcc package.

1.2 Related Documentation

Other documents that might be useful are:

- The **IPX−HOWTO**, which describes the details of configuring IPX protocol on Linux.
- The **Linux GCC HOWTO**, which covers how to set up the GNU C compiler and development libraries under Linux, and gives an overview of compiling, linking, running and debugging programs under it.
- The **Assembly HOWTO**, which describes how to program in assembly language using FREE programming tools, focusing on development for or from the Linux Operating System on the i386 platforms.
- The **Creating NLMs on Linux x86**, http://home.sch.bme.hu/~kereszttg/novell/howto/NLM−Linux−HOWTO.html, by Gabor Keresztvalvi <keresztg@mail.com>. His page describes the same thing as my HOWTO. I found Gabor's page ten days after releasing version 0.1 of this document :(.

1.3 Copying

2.Setting Up Your Linux Box and NetWare Server

You need to install and configure these things for NLM development:

- Linux box with configured IPX and NCPFS
- GNU C compiler (gcc) for elf−i386 (nearly all i386 Linux Distributions include it)
- GNU binutils with nlmconv(1) program
- My nlm−kit package
- Novell Developer Kit − NDK (include files and documentation)
- DOSemu (with rconsole) or X11 server for running NetWare Xconsole.
- ... and, finally, NetWare server :−)

You might also want to download the nlm−examples package from my FTP site: ftp://ftp.penguin.cz/pub/users/mhi/nlm/ or ftp://ftp.funet.fi/pub/mirrors/ftp.penguin.cz/pub/users/mhi/nlm/.

2.1 Novell NetWare Server

Let's start with the NetWare server. You can use NetWare versions 3.X, 4.X or 5.X. NetWare 5 (or 5.1) three user "demo" version can be ordered from Novell Inc. for a few dollars. Don't be confused with word "demo", it's fully functional NetWare, except that it is limited to three users. By the way, this license can be upgraded online, at no cost, to five users. You might also try asking your local Novell partner for demo CDs (they may be free).

2.2 Linux Box With IPX/NCPFS

You need to recompile your kernel with "The IPX protocol" and "NCP file system support" options enabled. Don't forget to say YES to "NDS authentication support" if you are using the NDS. Then you must configure the IPX protocol and mount your NetWare server volumes. Make sure that you have installed the ncpfs package! I use this script:

```
#!/bin/sh

ipx_interface delall
ipx_interface add -p eth0 802.2 120 # Frame Ethernet_802.2, ipx net num 120
insmod ncpfs                        # I have NCPfs compiled as module
ncpmount -U admin -S elf -P XYZ /nw # mount all ELF's volumes as /nw
```

For more information about configuring IPX and NCPFS, see the IPX−HOWTO.

2.3 GNU C Compiler

I think that all modern Linuxes for the Intel x86 include gcc, which generates ELF32/i386 object files. If you have an older Linux distribution, it may use the a.out format instead of ELF. If your Linux doesn't use ELF, get and install a newer gcc.

2.4 nlmconv(1) from GNU binutils

The nlmconv(1) utility links object files to the NLM format. It is a standard part of GNU binutils, but unfortunately it is not included in current distributions (RedHat, SuSE, Debian, ...). Get the binutils sources from ftp.gnu.org, and compile them, or simply use the pre−compiled nlmconv from my nlm−kit package.

2.5 The nlm−kit Package

My nlm−kit package is avaliable from ftp://ftp.penguin.cz/pub/users/mhi/nlm/. It contains necessary files for NLM development. Extract it and run `"make all"` and `"make install"`. It will create the directory `/usr/nwsdk/` and install all import files, object files and the nlmimp(1) utility.

2.6 Include Files and Documentation from the NDK

Getting the NDK is easy:

- Download it from http://developer.novell.com/ndk/.
- Order two NDK CDs from Novell.
- Get these CDs at no cost at BrainShare, Novell Developer Workshop or at other Novell Developer events.

You need these files from the NDK (all are available online at http://developer.novell.com/ndk/clib.htm:

- C language header and import files (cdrom:\files\download\clib.exe)
- C language API documentation (cdrom:\files\download\clib_doc.exe)
- C language samples (optional) (cdrom:\files\download\clib_sample.exe)

It's a pity that all the files mentioned above are InstallShield Win32 executables. You must find some Windows machine to extract them and then copy the include files to `/usr/nwsdk/include/` and documentation/samples to anywhere you want. The Novell License doesn't allow me to distribute include files or documentation with the nlm−kit.

Because the NDK include files don't work under Linux, you need to patch them manually by typing "make install−include" in the nlm−kit−X.Y/ directory.

2.7 Access to the NetWare Server (Xconsole or rconsole)

You can access the NetWare server console directly (keyboard and monitor), using rconsole.exe (from dosemu), or using telnetd.nlm/Xconsole (you need X server for this).

3.First Step: Hello world

As usual, we will start with the famous "Hello world" program. The source code for hello.nlm is available in the nlm−samples package. You can download it from ftp://ftp.penguin.cz/pub/users/mhi/nlm/.

3.1 hello.c − Source File

```
#define N_PLAT_NLM                              /* Define dest. platform */

#include <nwconio.h>                            /* ConsolePrintf */

int
main (int argc, char **argv)
{
  int i;

  ConsolePrintf ("\rHello world!\n\n");         /* print on system console */

  ConsolePrintf("Arguments:\n");                /* all arguments */
  for (i=0;i<argc;i++)
   ConsolePrintf("argv[%u]=\"%s\"\n",i, argv[i]);

  return 0;                                     /* exit NLM */
}
```

3.2 hello.def − NLM header file

```
#
# hello.def - NLM Header definition file for nlmconv(1)
# Copyright (c) 2000 Martin Hinner <mhi@penguin.cz>
#

# define startup object files
INPUT   hello.o
INPUT   /usr/nwsdk/lib/prelude.o            # clib startup code

# all imported functions and import lists
IMPORT @/usr/nwsdk/imports/clib.imp         # Functions in CLIB.NLM
IMPORT @/usr/nwsdk/imports/threads.imp      # Functions in THREADS.NLM

# NLM header...
```

```
OUTPUT  hello.nlm                                # output file
TYPE 0                                           # Ordinary NLM
VERSION 1,0,0                                    # Version 1.0
COPYRIGHT "Copyright (c) 2000 Martin Hinner <mhi@penguin.cz>" # (c) ...
DESCRIPTION "Simple 'Hello world' NLM module." # title of nlm
SCREENNAME "System Console"                      # Default screen name

MODULE CLIB,THREADS                              # req'd modules
```

3.3 Makefile

```
# makefile for "hello world" NLM

CC = gcc
CFLAGS = -Wall -O2 -g -I/usr/nwsdk/include/ -nostdinc -fno-builtin -fpack-struct

hello.nlm:      hello.o hello.def
        nlmconv --output-target=nlm32-i386 -T hello.def

hello.o:        hello.c
        $(CC) $(CFLAGS) -c hello.c
```

3.4 GCC problems

You must pass these arguments to the gcc:

- **−fno−builtin**: GCC's fast builtin functions sometimes cause server to abend, so we don't want to use them.
- **−nostdinc**: Only include files in /usr/nwsdk/include are valid for NLMs (don't forget to use also −I/usr/nwsdk/include).
- **−fpack−struct**: GCC's struct packing method is not valid for Novell NetWare, so we won't use it. Thanks to Gabor Keresztvalvi for this information.

3.5 Testing the Module

Copy hello.nlm to the SYS:\SYSTEM directory on your NetWare server. Then, on the system console, type "load hello.nlm". If everything went fine, you should see NLM version information, a copyright message and "Hello world".

4.NLM Header file

The NLM header file contains information for *nlmconv(1)*. Each line contains one option or directive; everything after "#" is comment. This chapter describes all options and directives.

This chapter is not yet finished, sorry.

4.1 AUTOUNLOAD

Syntax:

AUTOUNLOAD

4.2 CHECK

Syntax:

CHECK <check procedure name>

This directive specifies the function to be executed when the NLM is unloaded using the *UNLOAD* Server console command. If this function returns zero, the NLM can be unloaded, else the NLM is not ready to be unloaded.

Example:

```
CHECK CheckUnload
```

4.3 CODESTART

Syntax:

CODESTART <map file code start offset>

Map file start offset may be decimal or Xhex.

4.4 COPYRIGHT

Syntax:

COPYRIGHT ["Copyright string"]

The copyright string is displayed on the server console screen when the NLM is loaded. If this option is not used, no copyright information is displayed.

Example:

```
COPYRIGHT "Copyright (c) 1998 ABC Inc."
```

4.5 CUSTOM

Syntax:

CUSTOM <custom data file path>

4.6 DATASTART

Syntax:

DATASTART <map file data start offset>

Map file data start offset may be decimal or Xhex.

4.7 DATE

Syntax:

DATE <month, day, year>

4.8 DEBUG

Syntax:

DEBUG

This directive tells the nlmconv(1) to include debugging information in the NLM file.

Example:

```
DEBUG
```

4.9 DESCRIPTION

Syntax:

DESCRIPTION "NLM Description String"

4.10 EXIT

Syntax:

EXIT <exit procedure name>

4.11 EXPORT

Syntax:

EXPORT <symbol list>

EXPORT @<symbol list file>

4.12 FLAG_OFF

Syntax:

FLAG_OFF <decimal number>

4.13 FLAG_ON

Syntax:

FLAG_ON <decimal number>

4.14 HELP

Syntax:

HELP <help file path>

4.15 IMPORT

Syntax:

IMPORT <symbol list>

IMPORT @<symbol list file>

4.16 INPUT

Syntax:

INPUT <object file> [, <object file> [, ...]]

INPUT @<object list file>

This directive lists the input ELF (.o) object files that are to be linked. You can also list the object files in the list file, each object file on one line.

Example:

```
INPUT @objectfiles.txt
INPUT main.o
INPUT /usr/nwsdk/lib/prelude.o
```

4.17 MAP

Syntax:

MAP [map file name]

4.18 MESSAGES

Syntax:

MESSAGES <message file path>

4.19 MODULE

Syntax:

MODULE <autoload NLM list>

4.20 MULTIPLE

Syntax:

MULTIPLE

4.21 NAMELEN

Syntax:

NAMELEN <decimal number>

Default is 31. Zero is no limit.

4.22 OS_DOMAIN

Syntax:

OS_DOMAIN

4.23 OUTPUT

Syntax:

OUTPUT <target file name>

4.24 PATH

Syntax:

PATH [search path;...]

for following CUSTOM, HELP, INPUT, MESSAGES, SHARELIB, STAMPEDDATA and XDCDATA.

4.25 PSEUDOPREEMPTION

Syntax:

PSEUDOPREEMPTION

4.26 REENTRANT

Syntax:

REENTRANT

4.27 SCREENNAME

Syntax:

SCREENNAME "Initial Screen Name (CLIB)"

4.28 SHARELIB

Syntax:

SHARELIB <shared library path>

4.29 STACK

Syntax:

STACK <stack size>

4.30 STACKSIZE

Syntax:

STACKSIZE <stack size>

4.31 STAMPEDDATA

Syntax:

STAMPEDDATA "Stamp" <data file path>

Stamp is 8 char max.

4.32 START

Syntax:

START <start procedure name>

Default is _Prelude.

4.33 SYNCHRONIZE

Syntax:

SYNCHRONIZE

4.34 THREADNAME

Syntax:

THREADNAME "Initial Process Name (CLIB)"

4.35 TYPE

Syntax:

TYPE <version>

This directive specifies the format (NLM, LAN, DSK, NAM) of the NLM file to be generated. Valid values are:

- 0 – NLM
- 1 – LAN
- 2 – DSK
- 3 – NAM

Example:

```
TYPE 0
```

4.36 VERSION

Syntax:

VERSION <major version>, <minor version> [, <revision>]

The version information is displayed on the server system console when the NLM loads. The major and minor version numbers can be 0 – 99. The revision can be 0 – 26 ("a" – "z") and is optional. **The version directive is required.**

Example:

```
VERSION 1,5
```

4.37 XDCDATA

Syntax:

XDCDATA <XDC data file path>

5.Message files

A message file contains (as you guess) text messages generated by the NLM. You can create it using DOS programs MSGLIB.EXE and MSGMAKE.EXE. I don't know any similar utility for Unix. Sorry, you'll have to use dosemu or DOS machine :–(

6.Help Files

Help files contain help for use with the NWSNUT user interface library. There is no known Linux utility for creating help files. You must use the DOS program HELPLIB.EXE, which is available from Novell developer site.

7.XDC Data Files

XDC files are used by NetWare 5 (or SMP NetWare 4.x), and store information about symmetric multiprocessing (SMP). You will probably not need them. At least not now :−) Again, there is no Unix utility for creating XDC files, you will have to use the MPKXDC.EXE program (also available on the Novell developer site).

8.Header Files (.h)

(todo)

9.Import Files (.imp)

(todo)

9.1 Generating Import Files Using nlmimp(1)

Program nlmimp(1) is part of my nlm−kit package. (todo)
