Computer Science and Systems Analysis

Computer Science and Systems Analysis

Technical Reports

Miami University *Year* 1992

Librarian A Multi-User License Manager

Douglas Troy
Miami University, commons-admin@lib.muohio.edu

MIAMI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & SYSTEMS ANALYSIS

TECHNICAL REPORT: MU-SEAS-CSA-1992-010

Librarian: A Multi-User License Manager

Douglas A. Troy

School of Engineering & Applied Science | Oxford, Ohio 45056 | 513-529-5928

Librarian

A Multi-User License Manager

by

Douglas A. Troy
Systems Analysis Department
Miami University
Oxford, Ohio 45056

Working Paper #92-010 08/92

Librarian
A Multi-User License Manager

written by

The Applied Science Software Support Team
Miami University
Oxford, OH 45056

Contact: Douglas A. Troy
Systems Analysis Department
Miami University
Oxford, OH 45056
(513) 529-5934
E-mail: troyd@apsvax.aps.muohio.edu (Internet)
datroy@miamiu (Bitnet)

1.0 Introduction

A common problem in network administration is that there are often more workstations on a network than there are available licensed copies of the various software products. The purpose of the software described herein is to provide the tools necessary to manage and enforce software licensing agreements by insuring that no more that the licensed number of copies of any given package are in operation concurrently on a network. In addition, these tools collect usage statistics so that an administrator can determine where, when, and how many copies of each product are used. This type of information can prove invaluable in management and purchase decisions.

This software was designed to meet the following goals:

It should allow management of software products without modification to those products. The tools allow a centralized license manager to be queried before a product is run. Typically this is done through menu system or a shell script.

It should run on systems that support public domain network protocols such as TCP/IP and Remote Procedure Call. The software has been tested on IBM RS/6000 (AIX), IBM PCs running Sun's PC-NFS, NeXT Workstations, and DEC Vaxes (Ultrix).

It should support license management for IBM PC (MS-DOS) machines as well as UNIX machines, given a UNIX based file serve.

To accomplish these goals, the system was designed as an RPC client server application. In essence, each client that desires to run a licensed application must first "check out" that software from the server. When finished, clients "check in" the application. It is up to the client to follow the procedure. Also, tools are provided for the administrator to perform various management operations. The programs that constitute this package include:

Name	Usage	Description
check	client	Client program to check a software license in or out. Runs on clients or servers. DOS or UNIX.
libraria	server	Server program that maintains a real time database of the location and availability of currently running software products. This program also maintains a daily log of all operations. UNIX only.
pctime	client	An IBM PC program that gets the current date and time from the server and sets the PC's clock. DOS only.
libmaint	client or server	Allows the administrator to issue commands from a client to the server process. Commands include maintenance and query functions. DOS or UNIX.

Thus, before running a licensed program, a client would first issue a command like:

```
check myserver out lotus
```

This program sends an RPC request to the server program (running on the machine "myserver") to see if a copy of lotus is available. The check program waits for a reply and returns a status code to the operating system to indicate either "yes" or "no". Based on the result, the client can then proceed to run the application using whatever commands are required on the local network. Note that the check and libraria programs do not manage the software itself -- they only count check-in and check-out requests. After completing use of the application, a client then issues the command:

```
check myserver in lotus
```

To enforce the use of the check program, and administrator can embed the check commands into a menu manager or into UNIX scripts. Examples of these are shown in Section 7.

The license manager contains various safeguards to ensure the integrity of its database. If a client issues two check-outs without an intervening check-in, the librarian will automatically check the first product back in. (This means that check-in requests are not really necessary. However, since the libraria program logs all requests, explicit check-ins provide a good audit trail on each client's activities.) Another safe-guard is that the server program maintains an audit trail on disk of all client requests. If the server machine must be restarted while packages are checked-out (perhaps due to a crash) then upon start-up the server program will read the audit trail and return itself to its state prior to the crash.

For site licensed products, the administrator can set the number of licenses to zero to indicate "no limit." This is useful for tracking usage of site licensed products. The following sections describe the installation and operation of each of these programs.

2.0 Installation Procedures

The source code includes the following files:

```
check.c
libmaint.c
libraria.c
libraria.h
pctime.c
makefile
```

The programs check, pctime, and libmaint may be made for either the MS-DOS or UNIX environments. The program libraria can only be made for UNIX, as MS-DOS machines cannot act as an RPC server. For both environments, a library of Sun RPC functions and associated include files are required; This is standard on many UNIX systems; For MS-DOS, a library such as Sun's PC-NFS Toolkit must be purchased. The programs check, libmaint, and pctime have been compiled by the developers using Microsoft C.

Before making the programs, edit the file libraria.h and change the following macros to suit your site:

```
#define IDLE_MACHINE      /* Package name for nodes that have not
                             checked-out a package.  This could be the
                             operating system or menu manager. */
#define LICENSE_DB        /* file name for number of licenses */
#define ERRORLOGNAME      /* server error log file name */
#define MAX_PACKAGES      /* Max number of managed software packages */
#define MAX_CLIENTS       /* Max number of client machines */
```

2.1 Make for UNIX

The programs libraria, libmaint, and check can be made for UNIX. They can be simply compiled with libmaint.h and individually linked with the RPC library to produce the executables, or, optionally, the makefile, shown in Appendix 2, can be used. For example, to build libraria, with libraria.h in the current directory, use:

```
$ cc -o libraria libraria.c
```

Repeat the above command for check and libmaint.

2.2 Make for MS-DOS

To compile the MS-DOS version of the programs, adjust the environment to conform with the recommendations of your RPC toolkit supplier, and follow the directions of the vendor providing you C compiler. We use Microsoft C and Sun's PC-NFS Toolkit.

For example, to build check with libraria.h in the current directory using the Microsoft Quick-C compiler (small memory model), the Microsoft linker, and the PC-NFS Toolkit, use:

```
C> qcl -c check.c
C> link
Object Modules: check c:\toolkit\sprotrtn c:\toolkit\syp_rtns
Run File: check.exe
List File:
Libraries: slibtk
C>
```

Repeat the above command sequences for pctime and libmaint.

3.0 Libraria

The libraria program must run on the UNIX based server machine. It maintains the real time database of currently running applications and keeps a log of all requests that can be used later to produce usage reports.

3.1 Input

The input to the librarian is a file showing the number of licenses available for each product. The name of the file is given in libraria.h by the macro LICENSE_DB, the default name is software.lab. There is one line per product of the format:

number name

The following is an illustration:

```
0 AutoMenu
10 LOTUS
5 WP5
12 WS4
1 DB3P
10 SK
10 GV
12 QC
```

In this example, there are an unlimited number of copies of AutoMenu, 10 copies of LOTUS, 1 copy of D-Base III Professional, etc.

3.2 Startup

To initiate the librarian, place the following line in /etc/rc.local:

```
<your-path-name>librarian &
```

This will start the librarian at boot time. Note that the input file software.lab (or your version of LICENSE_DB) must be in the directory from which the libraria is started.

3.3 Output

Librarian generates a log, or audit file, for every transaction that it processes. The filenames of the files are of the following format:

Month_Date_Day.audit

The audit log shows the time of day (hh:mm:ss), the command that the server received, the IP address of the node issuing the command, and the package (if applicable) the command referred to. As these files will tend to accumulate, you may find it helpful to periodically delete older files. An example of the log file is shown below.

```
23:21:34 SET_MAX  000.000.000.000 AutoMenu          0 These lines
23:21:34 SET_MAX  000.000.000.000 LOTUS            10 show the
23:21:34 SET_MAX  000.000.000.000 WP5               5 initialization of
23:21:34 SET_MAX  000.000.000.000 WS4              12 the real time
23:21:34 SET_MAX  000.000.000.000 DB3P              1 database.
23:21:34 SET_MAX  000.000.000.000 SK               10
23:21:34 SET_MAX  000.000.000.000 GV               10
23:21:34 SET_MAX  000.000.000.000 QC               12
23:23:29 CHECK_OUT 134.053.002.003 LOTUS       A user checks out a license.
23:23:39 CHECK_IN 134.053.002.003 LOTUS        lotus is checked-in,
23:23:39 CHECK_OUT 124.053.002.003 AutoMenu    and the default package is
                                               automatically checked-out
23:24:11 CHECK_OUT 134.053.002.003 DB3P        Another license is checked
23:24:11 CHECK_IN 134.053.002.003 DB3P         in and out
23:24:11 CHECK_OUT 134.053.002.003 AutoMenu
23:25:20 QPACKAGE 134.053.003.002 LOTUS        Using libmaint, the manager
                                               checks the availability of
                                               lotus.
23:25:33 QNODES   134.053.003.002              Using libmaint, the manager at
                                               134.53.3.2 asked what nodes were
                                               known to the server.
23:29:21 SHUTDOWN 134.053.003.002              Using libmaint, the manager
                                               shuts down the server.
```

The librarian also generates an error log file. The name of the file is given by the macro ERRORLOGNAME in libraria.h, the default name is license.err.

4.0 Check

Check is the program that queries libraria to see if a package is available for use at a given time. It may be compiled under both UNIX and MS-DOS.

4.1 Input

This program has the following parameters:

```
check server-name <in|out> <software package>  OR
check server-name <software package> <-i|-o>
```

where <software package> is the name of a package in the software.lab (LICENSE_DB) file described in Section 3.1. The parameter "out" or "-o" is a request to check to see if a license is available for the software package; "in" or "-i" is a request to return a license to the available pool. The parameter "server-name" is the internet name for the server machine running the libraria program. When running check on a DOS machine under PC-NFS, the "server-name" must be present in the local HOSTS file.

4.2 Startup

To use check, run the program. For example, enter:

```
check myserver out LOTUS
```

If a license is available for the package, check will silently return to the operating system with an exit code of 0. If no licenses are available, check will display a message to the standard output and return 1 to the operating system.

4.3 Output

This program returns to the operating system a value of 0 upon successful request of permission to use a given package. If all licenses are in use, check will return a message telling the user to try later and also return 1 to the operating system.

5.0 Libmaint

Libmaint can be used to exercise all of the capabilities of the libraria program and to make modifications to the real time data base "on the fly", without shutting down the server or directly editing the config files. Libmaint allows the system administrator to adjust all the running characteristics of the server from any location, either on the server or on any client. It also provides the ability to query the server to determine what packages are checked out and what package each node is using.

5.1 Input

The libmaint options are as follows:

Usage: libmaint server-name <option>. The parameter server-name is the internet name of the machine running libraria -- for PCs this name must be present in the local HOSTS file. The parameter <option> is one of the following:

check <in|out>	same as the check program.
remove <node_number>	forbid an IP address from using the server.
query node <node_number>	see what license a node is using.
query package <package name>	see how many copies of a license are currently in use.
query time	find out the current time from the server.
query users	find out how many users are currently on the system.
query nodes	find out what nodes are currently using the system.
query license	display the number of licenses for each package.
set max <package_name> <software_maximum>	change the maximum number of licenses available. This overrides the LICENSE_DB file.
restart	restart libraria.
shutdown	clean up and terminate.

5.2 Startup

To start libmaint, use the command line format shown above. For example, to display the licenses available, use:

```
libmaint myserver query license
```

5.3 Output

The program will display text to the standard output appropriate to the command. See the Appendix 1 for examples. The values returned to the OS are:

Result	Meaning
0	no problems encountered.
1	node unknown.
-1	unexpected problem was encountered.

6.0 PCtime

PCtime is a client program that allows an IBM PC to query the time from the server and set its internal clock accordingly.

6.1 Input

No input is required.

6.2 Startup

To invoke the pctime program, use the command:

```
pctime server-name
```

where "server-name" is the internet name for the server machine defined in the local HOSTS file. This command would reside typically in the AUTOEXEC.BAT file.

6.3 Output

Pctime displays the current time as shown on the servers system clock as well as setting the PC's internal clock.

7.0 Example Use of Check

7.1 Use with DOS

To use check under DOS, two approaches can be followed:

1. Write a DOS batch file for each licensed application, or
2. Use a menu system such as AutoMenu that allows use of DOS batch commands within the menu system.

In either case, the idea is to use the DOS batch file language to first call check to see if a licensed copy is available and then to conditionally load the application only if the return from check is affirmative. An example batch file, which executes the program gwbasic from the network server "apsrisc" if a license is available, is shown below:

```
# Batch file to run gwbasic
check apsrisc out gwbasic
if errorlevel 1 goto DONE
net use r: \\server\appls\langs\basic
r:
gwbasic
c:
net use r: /d
check apsrisc in gwbasic
:DONE
```

Another approach is to build commands similar to the above into a menu system. The program AutoMenu supports the use of the above batch file commands within the menu system itself, eliminating the need for a separate batch file for every licensed software package.

7.2 Use with Unix

To use check under Unix, a shell script can be built using the appropriate Unix shell language. An example Bourne Shell script is illustrated below.

```
#!/bin/sh
# Shell script to run netlab
check apsrisc out netlab
if test $? = 0
then
        netlab
        check apsrisc in netlab
fi
```

Appendix 1 -- Libmaint Commands

Below is session using all the commands available in libmaint. Comments are in parentheses.

```
$ libmaint                              (What can I do?)
Usage: libmaint <server> <option>, where <option> is:
                         check in  <package_name>
                         check out <package_name>
                         query node    <node_number>
                         query package <package_name>
                         query time
                         query users
                         query nodes
                         query license
                         remove        <node_number>
                         set max <package_name> <software_maximum>
                         restart
                         shutdown
$ libmaint apsrisc check out LOTUS       (Check out a license.)
Librarian Result is 0
$ libmaint apsrisc check in LOTUS        (Return the license.)
Librarian Result is 0
$ libmaint apsrisc remove 134.53.2.2     (Remove a node.)
Librarian Result is 0
$ libmaint apsrisc query node 127.0.0.1  (What license has a node?)
Node Number     : 127.0.0.1
Software Package: AutoMenu               (The default program for an idle node.)
$ libmaint apsrisc query package AutoMenu      (How many licenses in use?)
Software Package: AutoMenu
Copies Available:  0 this means infinite copies allowed
Copies in Use   :  1
$ libmaint apsrisc query time            (What time has the server?)
Current date is Wed 11-14-1990
Current time is 21:26:57.00
$ libmaint apsrisc query users           (What nodes have requested licenses?)
Machines Registered:  1
Machines     in Use:  0
Machines       Idle:  1
$ libmaint apsrisc query nodes           (What is each nodes doing?)
127.000.000.001 AutoMenu
$ libmaint apsrisc query license         (What licenses managed, How many in use?)
AutoMenu          0   1
LOTUS            10   0
WP5               5   0
WS4              12   0
DB3P              1   0
SK               10   0
GV               10   0
QC               12   0
$ libmaint apsrisc set max LOTUS 3       (Change number of licenses allowed.)
Librarian Result is 0
$ libmaint apsrisc restart               (Read in old log file & pick up where we
                                          left off.)
Librarian Result is 0
$ libmaint apsrisc shutdown              (Terminate libraria program.)
$
```

Appendix 2 -- Make file for UNIX

```
## Makefile for librarian
##
## Tools we need to use
RM = /bin/rm -f
CC = /bin/cc
CP = /bin/cp
INSTALL = /usr/bin/install -c
##
##
## Locations of final files
DESTDIR = /usr/local/librarian
##
##      -g = debug
##      -O = optimize (precludes -g)
CFLAGS= -O
##
## LIBRARIES TO LINK TO:
##
##
SRC1 =    librarian.c
SRC2 =    check.c
SRC3 =    libmaint.c
##
OBJECT1= librarian.o
OBJECT2= check.o
OBJECT3= libmaint.o
OBJECT4= libtime.o
##
PROGRAM1 =     librarian
PROGRAM2 =     check
PROGRAM3 =     libmaint
PROGRAMS =     $(PROGRAM1) $(PROGRAM2) $(PROGRAM3)
##
##
JUNK =
   $(PROGRAMS) *.audit *.o
OBJS =
   $(OBJECT1) $(OBJECT2) $(OBJECT3)
##
## compile
##
all::     $(PROGRAMS)
##
##
install:;  $(INSTALL) $(PROGRAMS) $(DESTDIR)
##
##
##
librarian:     $(OBJECTS1)
   $(CC) $(CFLAGS) -o $(PROGRAM1) $(SRC1)
#  $(RM) $(OBJECT1)
```

```
##
##
check:    $(OBJECTS2)
   $(CC) $(CFLAGS) -o $(PROGRAM2) $(SRC2)
#  $(RM) $(OBJECT2)
##
##
libmaint:      $(OBJECTS3)
   $(CC) $(CFLAGS) -o $(PROGRAM3) $(SRC3)
## $(RM) $(OBJECT3)
##
libtime: $(OBJECTS4)
   $(CC) $(CFLAGS) -o $(PROGRAM4) $(SRC4)
## $(RM) $(OBJECT4)

##
##
clean::
   $(RM) $(JUNK)
##
##
```

Edit the makefile to reflect where you would like the final package to go. Also check to see if the tools are listed where they reside on your system. The Makefile says that the -g and -O compiler options are mutually exclusive. That is only true if you are not using gcc. "csh> make all" will compile all the source. "csh> make install" will put the files where they belong. "csh> make clean" will remove the object files from the directory where they were compiled.

Site specific macros that should be edited are contained in Libraria.h and are listed below:

```
#define IDLE_MACHINE      /* Package name for notes that have checked-out a
                             package.   This could be the operating system or menu
                             manager. */
#define LICENSE_DB        /* file name for number of licenses */
#define ERRORLOGNAME      /* server error log file name */
#define MAX_PACKAGES      /* Max number of managed software packages */
#define MAX_CLIENTS       /* Max number of client machines */
```