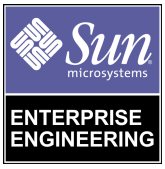
Solaris Resource Manager™: Resource Assignment

By Richard McDougall - Enterprise Engineering

Sun BluePrints™ OnLine - August 1999



http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 806-4638-10
Revision 01, August 1999

Solaris Resource Manager™: Resource Assignment

Quite a few questions have been raised about how the resource description ‘lnodes’ are attached to users and whether they are transferred when changing userids with the `su` and related commands. This article explains how users are assigned resource lnodes and under what circumstances they change to ensure that resource limits are allocated correctly.

How lnodes Are Assigned

Solaris Resource Manager™ uses the concept of limit-nodes or lnodes to assign, control, and account for resources. The default allocation of lnodes is by userid, whereby each user is assigned a unique lnode as they log into the system. The default allocation of a lnode to each user allows the control of resources that are allocated on a per user basis.

The Pluggable Authentication Modules (PAM) controls the assignment of users to lnodes during the log-in process. Whenever a user requests an operation that involves changing or setting the user’s identity (such as logging into the system, invoking an ’`r`’ command such as `rcp` or `rsh`, using `ftp`, or using `su`), a set of configurable modules provide authentication, account management, credentials management, and session management. Solaris Resource Manager provides a module for login accounting, and modifying the behavior of `su`. The `pam_srm(5SRM)` manual pages include a full discussion of how Solaris Resource Manager uses PAM for account and session management.

By default, the lnode assignment is done when:

- A user logs into the system via `/bin/login`, `in.ftpd` or any other PAM enabled login points.
- When `su` is invoked to change users (only true for some transitions)
- When the `setuid()` system call is invoked

Although lnodes are assigned to users by default, there is no strong binding between the userid a process runs as and the lnode it is attached to. The user-lnode default mapping is simply that--just a default. There are situations where we want to assign and run processes as different lnodes than the user that the process runs as, and in this case we can make use of the srmuser and su commands to accomplish this. For example, if you want to start an entire database instance from root, but have it attached to an lnode of 'database', you can use the `srmuser` command to launch the database start script as root with an alternative lnode attachment.

```
# srmuser database /export/database/bin/startdb.sh
```

When you use the `srmuser` command, the initial shell is attached to the database lnode, and the script runs as the root userid. Each process that the `startdb` script forks inherits and stays attached to the database lnode.

Attachment Across Set userid Binaries

Note that lnode assignment does not change when a `setuserid` binary is executed. For example, a binary may be owned by userid "database", and if user "fred" executes this binary then the process is run as userid "database", but the process will remain attached to "fred's" lnode, and resource allocation and accounting will be done on that basis. This happens because the setuid binary only changes the effective userid of the process, and a change in lnode only occurs when the real userid is changed with the `setuid` system call.

Changing userids with the `su` Command

The `/bin/su` command can be used to change the userid of a session, or to run a process as another user. The su command uses the setuid system call to change the real userid of a process, which can change the lnode that processes are assigned to. However the `su` implementation does not change the lnode in all cases. Table 1 on page 3 shows under what circumstances the lnode is changed during a change of users with the `su` command.

Table 1: `su` **lnode Assignment Rules**

`su` **from**	`su` **to**	**change lnode?**
root	non-root user	yes
non-root user	root	no
non-root user X	non-root user Y	yes

Jobs Launched by root as Different Users

The operating system can start processes as non-root users from the `cron` daemon and by the `inet` daemon. Processes started this way begin as a fork from the root user, and before the processes are executed the calling process switches to the target user by using the `setuid` system call. The process is started with the lnode of the user the processes are started under.

```
# ps -aef |grep cron
root   201     1  0 10:32:54 ?        0:00 /usr/sbin/cron
```

For example, `cron` may launch a job from user “fred’s” `crontab` file, and although the `crontab` daemon that starts the process is root, the `setuid` system call changes both userid and lnode to the user of the `crontab` file before the process is started.

```
# truss -f -p 201
1347:   fork()          (returning as child ...)        = 201
1347:   setuid(36413)                                   = 0
```

In the example above, the `cron` daemon changed users with the setuid system call before the process was started. This forces a change of lnodes to the target user.

Processes Started with the inet Daemon

The `inet` daemon can also start processes as users other than root. These processes are started in the same way as with `cron`, by a fork and `setuid`. All processes started with `inet` are assigned to the correct lnode, corresponding with the user specified in the user field of the `inetd` configuration.

Starting Processes for Users Without lnodes

One last case that justifies some attention is when processes are started for a user which has no lnode. Since lnodes are created the first time a user logs in, this can occur if `cron`, `su` or another process that calls the setuid system call to change users *before* that user has logged into the system. In such a case, the processes are attached to a special lnode assigned to the 'lost' user, which is created when Solaris Resource Manager is installed

Summary

In summary, processes are only assigned when the setuid system call is executed. This occurs either directly (`su`, `inetd`) or indirectly by the `su` command.

Author's Bio: Richard Mc Dougall

Richard has over 11 years of UNIX experience including application design, kernel development and performance analysis, and specializes in operating system tools and architecture.