Shared Memory

The SunSoft Developer Engineering team is dedicated to helping developers migrate their software applications to SunSoft products, and will provide the answers to your questions. Please submit your queries to us via E-Mail at dev_eng@Sun.COM. We will attempt to answer all questions posted to us, but we may not have room to print all responses.

How do I use shared memory under Solaris?

There are two kinds of shared memory: objects opened with mmap(2) and System V Shared Memory. Developers should always use the mmap(2) system call whenever possible.

The documentation reflects this preference, as stated in the *System V Interprocess Communication Facilities* document, page 3-35: “In the SunOS operating system, the most efficient method for implementing shared memory applications is to rely on native virtual memory management and the mmap(2) system call.” However, those vendors that insist on using the System V Shared Memory, and want to change the default values, need to be familiar with the <sys/shm.h>:shminfo structure. The default values can be tuned using the /etc/system file syntax. The system default values are:

shminfo_shmmax = 1048576 /* max shared memory segment size */

shminfo_shmmin = 1 /* min shared memory segment size */

shminfo_shmmni = 100 /* # of shared memory identifiers */

shminfo_shmseg = 6 /* segments per process */

The shared memory facility is a loadable module called shmsys. For example, to double the maximum shared memory segment size, add the following line to /etc/system file, and then reboot:

```
set shmsys:shminfo_shmmax=0x200000
```

where `shmsys` is the module name (1048576 * 2)

Unfortunately, the system does not provide any utility program to print these values. The ipcs (1) command is the only utility which prints information about “active” shared memory, message queues, and semaphores. However, these values can easily be obtained from the kernel memory. The attached program is an example of how one can extract this information using the “kvm” library. The shared memory module is automatically loaded by the system when shmget (2) is first executed. If no shared memory program is running, the module has not been loaded yet. You can make the module available by using the modload(1M) command:

```
# modload -p sys/shmsys
```

(See separate program “shminfo.c” which extracts shared memory information from /dev/kvm)