Q: In my application, I invoke other commands and programs by using **system**() or other relatives like **popen**(), **execl**() . Sometimes when debugging my application, I run into this error message:

```
        sh: privileges disabled because of outstanding IPC access to task
```

What does it mean and what can I do about it?

A: This has to do with an unfortunate interaction between gdb and setuid program execution. When gdb is debugging a process, it owns the exception ports of that process. When that process forks a child process, gdb would own the exception ports of that child process as well. Because of security issues, the kernel disallows gdb from owning the exceptions ports of a child process that is setuid.   When you attempt this, the kernel generates the privileges error message and the **system**() call fails.

There will be no conflict outside the debugger and you can run gdb as root as a workaround for debugging.

QA849

Valid for 1.0, 2.0, 3.0