
Subject: Instructions of how to make testing easy
Posted by [Pavel Emelianov](#) on Thu, 24 May 2007 13:19:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

That's the program I used for testing. It creates a new session, chroots to new root, clones the namespace, mounts proc and launches the sshd to keep track of the terminals.

The new root I prepared was bind-mounted /lib, /bin, /usr etc directories, copied /dev devices with devpts mounted inside and empty /var (for sshd) and /proc (for new mount).

After these preparations I launched this enter program and then used ssh to get into the namespace.

Hope this will help.

The patches introduced was then tested with some mportaint tests from ltp testsuite in 4 ways:

- 1 kernel w/o the patch
- 2 kernel with CONFIG_PID_NS=n
- 3 kernel with namespaces in init namespace
- 4 kernel with namespaces in subnamespace

The results coincided.

```
#include <unistd.h>
#include <sys/wait.h>
#include <errno.h>
#include <linux/unistd.h>
```

```
#ifndef __NR_unshare
#define __NR_unshare 310
#endif
```

```
_syscall1(int, unshare, int, flags)
```

```
#define CLONE_NEWPIDS 0x10000000
#define ROOT_DIR "./new_root"
```

```
int main(void)
{
    int pid;
    int status;
```

```
    pid = fork();
    if (pid < 0) {
```

```

perror("Can't fork\n");
return 1;
}

if (pid > 0) {
    if (waitpid(pid, &status, 0) < 0) {
        perror("Can't wait kid\n");
        return 2;
    }

    if (WIFEXITED(status))
        printf("%d exited with %d/%d\n", pid,
            WEXITSTATUS(status) & 0xf,
            WEXITSTATUS(status) >> 3);
    else if (WIFSIGNALED(status))
        printf("%d signalled with %d\n", pid, WTERMSIG(status));
    else
        printf("Some shit happened with %d\n", pid);
    return 0;
}

printf("Set new sid\n");
if (setsid() < 0)
    return (errno << 4) + 0;

printf("Unshare\n");
if (unshare(CLONE_NEWPIDS) < 0)
    return (errno << 4) + 2;

printf("Chroot\n");
if (chroot(ROOT_DIR) < 0)
    return (errno << 4) + 1;

printf("Mount proc\n");
if (mount("none", "/proc", "proc", 0, NULL) < 0)
    return (errno << 4) + 3;

printf("Launching sshd\n");
if (fork() == 0) {
    execl("/usr/sbin/sshd", "/usr/sbin/sshd", "-p", "2202", NULL);
    return (errno << 4) + 3;
}

/* Never exit... Bad init */
while (1) {
    if (wait(NULL) < 0)
        sleep(1);
}

```

```
return 0;  
}
```
