

My very dirty solution to fix it

fs_simfs_ro.c:

```
#include <stdlib.h>
#include <unistd.h>
#include <sys/mount.h>
#include <sys/vfs.h>
#include <errno.h>
#include <stdio.h>
#include <string.h>
```

```
#include "types.h"
#include "fs.h"
#include "logger.h"
#include "vzerror.h"
```

```
int vz_mount(fs_param *fs, int remount)
{
    int mntopt = 0;

    if (fs->noatime == YES)
        mntopt |= MS_NOATIME;
    if (remount)
        mntopt |= MS_REMOUNT;

    mntopt |= MS_RDONLY;

    logger(2, 0, "Mounting root: %s %s", fs->root, fs->private);
    if (mount(fs->private, fs->root, "simfs", mntopt,
        remount ? "" : fs->private) < 0)
    {
        logger(-1, errno, "Can't mount: %s %s", fs->root, fs->private);
        if (errno == ENODEV)
            logger(-1, errno, "Kernel lacks simfs support. Please "
                "compile it in, or load simfs module.");
        return VZ_FS_CANTMOUNT;
    }
    return 0;
}
```

```
gcc -I../include/ -Wall -O2 -fpic -shared -ldl -o libfs_simfs_ro.so fs_simfs_ro.c
```

New /usr/sbin/vzctl wrapper

```
#!/bin/sh
```

```
if [ x$1 = xstart_ro ]; then
    LD_PRELOAD=/usr/lib/libfs_simfs_ro.so
    export LD_PRELOAD
    shift
    /usr/sbin/vzctl.orig start $@
else
    /usr/sbin/vzctl.orig $@
fi
```

Maybe it'll help somebody
