

linux chroot 命令

chroot，即 change root directory (更改 root 目录)。在 linux 系统中，系统默认的目录结构都是以 /，即以根 (root) 开始的。而在使用 chroot 之后，系统的目录结构将以指定的位置作为 / 位置。

基本语法

chroot NEWROOT [COMMAND [ARG]...]

具体用法请参考本文的 demo。

为什么要使用 chroot 命令

增加了系统的安全性，限制了用户的权力：

在经过 chroot 之后，在新根下将访问不到旧系统的根目录结构和文件，这样就增强了系统的安全性。一般会在用户登录前应用 chroot，把用户的访问能力控制在一定的范围之内。

建立一个与原系统隔离的系统目录结构，方便用户的开发：

使用 chroot 后，系统读取的是新根下的目录和文件，这是一个与原系统根下文件不相关的目录结构。在这个新的环境中，可以用来测试软件的静态编译以及一些与系统不相关的独立开发。

切换系统的根目录位置，引导 Linux 系统启动以及急救系统等：

chroot 的作用就是切换系统的根位置，而这个作用最为明显的是在系统初始引导磁盘的处理过程中使用，从初始 RAM 磁盘 (initrd) 切换系统的根位置并执行真正的 init，本文的最后一个 demo 会详细的介绍这种用法。

通过 chroot 运行 busybox 工具

busybox 包含了丰富的工具，我们可以把这些工具放置在一个目录下，然后通过 chroot 构造出一个 mini 系统。简单起见我们直接使用 docker 的 busybox 镜像打包的文件系统。先在当前目录下创建一个目录 rootfs：

```
$ mkdir rootfs
```

然后把 busybox 镜像中的文件释放到这个目录中：

```
$ (docker export $(docker create busybox) | tar -C rootfs -xvf -)
```

通过 ls 命令查看 rootfs 文件夹下的内容：

```
$ ls rootfs
```

```
nick@tiger:/tmp$ ls rootfs/
bin dev etc home proc root sys tmp usr var
```

万事俱备，让我们开始吧！

执行 chroot 后的 ls 命令

```
$ sudo chroot rootfs /bin/ls
```

```
nick@tiger:/tmp$ sudo chroot rootfs /bin/ls
bin dev etc home proc root sys tmp usr var
```

虽然输出结果与刚才执行的 ls rootfs 命令形同，但是这次运行的命令却是 rootfs/bin/ls。

运行 chroot 后的 pwd 命令

```
$ sudo chroot rootfs /bin/pwd
```

```
nick@tiger:/tmp$ sudo chroot rootfs /bin/pwd
/
```

哈，pwd 命令真把 rootfs 目录当根目录了！

不带命令执行 chroot

```
$ sudo chroot rootfs
```

```
nick@tiger:/tmp$ sudo chroot rootfs
chroot: failed to run command '/bin/bash': No such file or directory
```

公告

昵称：sparkdev  
园龄：5年8个月  
荣誉：推荐博客  
粉丝：976  
关注：32  
+加关注

| 2022年1月 |    |    |    |    |    |  |  |  |  |  |  |  |  |
|---------|----|----|----|----|----|--|--|--|--|--|--|--|--|
| 日       | 一  | 二  | 三  | 四  | 五  |  |  |  |  |  |  |  |  |
| 26      | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |
| 2       | 3  | 4  | 5  | 6  | 7  |  |  |  |  |  |  |  |  |
| 9       | 10 | 11 | 12 | 13 | 14 |  |  |  |  |  |  |  |  |
| 16      | 17 | 18 | 19 | 20 | 21 |  |  |  |  |  |  |  |  |
| 23      | 24 | 25 | 26 | 27 | 28 |  |  |  |  |  |  |  |  |
| 30      | 31 | 1  | 2  | 3  | 4  |  |  |  |  |  |  |  |  |

搜索

找找看

谷歌搜索

最新随笔

- 1.创建 SysV 风格的 linux daemon 程序
- 2.Linux session(会话)
- 3.Linux job control
- 4.Linux 伪终端(pty)
- 5.Linux 终端(TTY)
- 6.Linux Capabilities 简介
- 7.用什么监控我们的容器?
- 8.Ubuntu Server：自动更新
- 9.Ubuntu：apt 命令
- 10.Ubuntu：apt-get 命令

我的标签

- Linux(80)
- docker(38)
- Azure(28)
- Golang(16)
- PowerShell(15)
- jenkins(12)
- CI/CD(10)
- ubuntu(10)
- teamcity(9)
- log(9)
- 更多

这次出错了，因为找不到 /bin/bash。我们知道 busybox 中是不包含 bash 的，但是 chroot 命令为什么会找 bash 命令呢？原来，如果不给 chroot 指定执行的命令，默认它会执行 '\$(SHELL) -i'，而我的系统中 \$(SHELL) 为 /bin/bash。既然 busybox 中没有 bash，我们只好指定 /bin/sh 来执行 shell 了。

```
$ sudo chroot rootfs /bin/sh

nick@tiger:/tmp$ sudo chroot rootfs /bin/sh
/ # echo $$
46644
```

运行 sh 是没有问题的，并且我们打印出了当前进程的 PID。

## 检查程序是否运行在 chroot 环境下

虽然我们做了好几个实验，但是肯定会有朋友心存疑问，怎么能证明我们运行的命令就是在 chroot 目录后的路径中呢？

其实，我们可以通过 /proc 目录下的文件检查进程的中的根目录，比如我们可以通过下面的代码检查上面运行的 /bin/sh 命令的根目录(请在另外一个 shell 中执行)：

```
$ pid=$(pidof -s sh)
$ sudo ls -ld /proc/$pid/root

nick@tiger:/tmp$ pid=$(pidof -s sh)
nick@tiger:/tmp$ sudo ls -ld /proc/$pid/root
lrwxrwxrwx 1 root root 0 Mar  5 18:45 /proc/46644/root -> /tmp/rootfs
```

输出中的内容明确的指出 PID 为 46644 的进程的根目录被映射到了 /tmp/rootfs 目录。

## 通过代码理解 chroot 命令

下面我们尝试自己实现一个 chroot 程序，代码中涉及到两个函数，分别是 chroot() 函数和 chdir() 函数，其实真正的 chroot 命令也是通过调用它们实现的：

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    if(argc<2){
        printf("Usage: chroot NEWROOT [COMMAND...] \n");
        return 1;
    }

    if(chroot(argv[1])) {
        perror("chroot");
        return 1;
    }

    if(chdir("/")) {
        perror("chdir");
        return 1;
    }

    if(argc == 2) {
        // hardcode /bin/sh for my busybox tools.
        argv[0] = (char *) "/bin/sh";

        argv[1] = (char *) "-i";
        argv[2] = NULL;
    } else {
        argv += 2;
    }

    execvp (argv[0], argv);
    printf("chroot: cannot run command '%s'\n", *argv);

    return 0;
}
```

把上面的代码保存到文件 mychroot.c 文件中，并执行下面的命令进行编译：

```
$ gcc -Wall mychroot.c -o mychroot
```

mychroot 的用法和 chroot 基本相同：

```
$ sudo ./mychroot ./rootfs
```

积分与排名

积分 - 703313  
排名 - 551

随笔分类 (346)

AI(2)  
Ansible(3)  
Azure(28)  
Bash(8)  
C#(9)  
cgrouops(2)  
CI/CD(20)  
DevOps(16)  
Docker(39)  
ElasticSearch(1)  
elk(9)  
Git(4)  
Golang(16)  
https(1)  
Jenkins(12)  
更多

随笔档案 (231)

2020年4月(1)  
2020年1月(1)  
2019年12月(1)  
2019年9月(2)  
2019年8月(7)  
2019年7月(7)  
2019年6月(7)  
2019年5月(7)  
2019年4月(7)  
2019年3月(7)  
2019年2月(7)  
2019年1月(7)  
2018年12月(7)  
2018年11月(7)  
2018年10月(7)  
更多

阅读排行榜

1. Dockerfile 中的 COPY 与 ADD 命令(229443)
2. SSH 远程执行任务(133325)
3. linux sudo 命令(111763)
4. Docker: 限制容器可用的 CPU(1074)
5. Docker: 限制容器可用的内存(1033)
6. Windows 支持 OpenSSH 了! (1517)
7. Docker Compose 引用环境变量(1021)
8. linux useradd 命令基本用法(838)
9. Linux mount 命令(80165)
10. Dockerfile 中的 CMD 与 ENTRYPOINT(72958)

评论排行榜

```
n1ck@ttigger:/tmp$ sudo ./mychroot rootfs
/ # ls
bin  dev  etc  home  proc  root  sys  tmp  usr  var
```

特别之处是我们的 mychroot 在没有传递命令的情况下执行了 /bin/sh，原因当然是为了支持我们的 busybox 工具集，笔者在代码中 hardcoded 了默认的 shell：

```
argv[0] = (char *)"/bin/sh";
```

从代码中我们也可以看到，实现 chroot 命令的核心逻辑其实并不复杂。

## 实例：通过 chroot 重新设置 root 密码

忘记了 root 密码该怎么办？接下来的 demo 将演示如何通过 chroot 命令重新设置 centos7 中被忘记了的 root 密码。

systemd 的管理机制中，rescue 模式和 emergency 模式是无法直接取得 root 权限的，需要使用 root 密码才能进入 rescue 和 emergency 环境。所以我们需要通过其他方式来设置 root 密码。我们可以为内核的启动指定 "rd.break" 参数，从而让系统在启动的早期停下来，此时我们可以通过使用 root 权限并结合 chroot 命令完成设置 root 密码的操作。下面我们一起来看具体的操作过程。

在系统启动过程中进入开机菜单时按下字母键 e 进入开机菜单的编辑模式：

```
CentOS Linux (3.10.0-693.el7.x86_64) 7 (Core)
CentOS Linux (0-rescue-326bc2ec00374c49a411927ffdac0b68) 7 (Core)

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
The selected entry will be started automatically in 4s.
```

这就是系统的开机菜单，按下 e 后进入编辑界面：

```
insmod xfs
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 31b1a2ce-8\
d3c-40a9-af2b-08e9878a02db
else
  search --no-floppy --fs-uuid --set=root 31b1a2ce-8d3c-40a9-af2b-08e9\
878a02db
fi
linux16 /vmlinuz-3.10.0-693.el7.x86_64 root=/dev/mapper/centos-root ro\
crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet LANG=\
en_US.UTF-8
initrd16 /initramfs-3.10.0-693.el7.x86_64.img
```

找到以 "linux16 /vmlinuz-" 开头的行。如果默认没有看到该行，需要按向下键把它滚动出来。

然后定位到该行结尾处，输入一个空格和字符串 "rd.break"，如下图所示：

```
linux16 /vmlinuz-3.10.0-693.el7.x86_64 root=/dev/mapper/centos-root ro\
crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet LANG=\
en_US.UTF-8 rd.break
```

接着按下 ctrl + x 以该设置继续启动，启动过程中操作系统会停下来，这是系统启动过程中的一个非常早的时间点：

```
[ 1.107596] sd 2:0:0:0: [sda] Assuming drive cache: write through
Generating "/run/initramfs/rdsosreport.txt"
[ 1.534315] blk_update_request: I/O error, dev fd0, sector 0
[ 1.597312] blk_update_request: I/O error, dev fd0, sector 0

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

switch_root:/#
```

所以系统的根目录还挂载在 RAM disk 上(就是内存中的一个文件系统)，我们可以通过 mount 命令检查系统当前挂载的文件系统，下面是我们比较关心的两条：

```
switch_root:/# mount
rootfs on / type rootfs (rw)
/dev/mapper/centos-root on /sysroot type xfs (ro,relatime,attr2,inode64,noquota)
```

上图中 mount 命令输出的第一行说明此时的根目录在一个 RAM disk 中，即 rootfs。

图中输出的第二行说明我们的文件系统此时被挂载到了 /sysroot 目录，并且是只读的模式：

```
/dev/mapper/centos-root on /sysroot type xfs (ro,relatime,attr2,inode64,noquota)
```

而在我们正常登陆系统的情况下，系统根目录的挂载情况如下：

```
/dev/mapper/centos-root on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

1. Windows 支持 OpenSSH 了！(4)
2. Docker Machine 详解(37)
3. Docker Machine 简介(34)
4. 用 Docker Machine 创建 Azure 拟主机(29)
5. SSH 远程执行任务(29)
6. C# 创建压缩文件(28)
7. 局域网内部署 Docker Registry(2)
8. Python 操作 Azure Blob Storage(4)
9. C# BackgroundWorker 详解(23)
10. PowerShell 远程执行任务(22)

推荐排行榜

1. SSH 远程执行任务(61)
2. Windows 支持 OpenSSH 了！(5)
3. Dockerfile 中的 CMD 与 ENTRYPOINT(52)
4. Docker Machine 详解(43)
5. 从 docker 到 runC(41)

最新评论

1. Re:Golang 入门：切片(slice)  
在切片的容量小于 1000 个元素时，是会成倍地增加容量。一旦元素个数超过 1000，容量的增长因子会设为 1.2。目前里程碑是 256。  
--博客猿马甲
2. Re:Golang 入门：切片(slice)  
@零-壹 make 的时候，不是可以设置参数 3，指定 cap 吗？...  
--博客猿马甲
3. Re:Linux AUFS 文件系统  
不行啊 ..  
mount: 未知的文件系统类型 "aufs" ..  
--CanntBelieve
4. Re:Golang 入门：切片(slice)  
图文并茂，博主用心了，谢谢分享，我很有帮助。这边问下您可以把您的文章转载到 ApiPost 博客中展示吗，当然我们会标明出处  
--CodeNong
5. Re:linux free 命令  
您好，想请问下这个 free 怎么安装呢？应该不是系统自带的吧，百度有说先装 fish，再安装 free，但是老报错。  
--商纬以

该时间点的最大优势是我们具有 root 权限！所以让我们开始设置新的 root 密码吧。

先通过下面的命令把 /sysroot 重新挂载为可读写的模式：

```
switch_root:/# mount -o remount,rw /sysroot
```

然后用下面 chroot 命令把根目录切换到我们原来的环境中：

```
switch_root:/# chroot /sysroot
```

此时可以理解为：我们以 root 权限登录了原来的系统，修改密码就很容易了！用下面的命令为 root 用户设置新的密码：

```
sh-4.2# echo "new_root_pw" | passwd --stdin root
```

接下来还要处理 SELinux 相关的问题。由于当前的环境中 SELinux 并未启动，所以我们对文件的修改可能造成文件的 context 不正确。为了确保开机时重新设定 SELinux context，必须在根目录下添加隐藏文件 .autorelabel：

```
sh-4.2# touch /.autorelabel
```

最后从 chroot 中退出，并重启系统：

```
sh-4.2# exit
switch_root:/# reboot
```

重新进入登陆界面时就可以使用刚才设置的密码以 root 登陆了！


## 总结

chroot 是一个很有意思的命令，我们可以用它来简单的实现文件系统的隔离。但在一个容器技术繁荣的时代，用 chroot 来进行资源的隔离实在是 low 了点。所以 chroot 的主要用途还是集中在系统救援、维护等一些特殊的场景中。

参考：

理解 chroot

Linux – RedHat7 / CentOS 7 忘记root密码修改



作者: sparkdev  
出处: <http://www.cnblogs.com/sparkdev/>  
本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类: [Linux](#)  
标签: [chroot](#), [Linux](#)

好文置顶

关注我

收藏该文







sparkdev  
关注 - 32  
粉丝 - 976

推荐博客  
[+加关注](#)

« 上一篇：[Dockerfile 中的 multi-stage\(多阶段构建\)](#)  
» 下一篇：[减小容器镜像的三板斧](#)


8

0

 推荐  反对

posted @ 2018-03-15 10:32 sparkdev 阅读(22232) 评论(4) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

编辑推荐：

- 技术部如何做复盘——“年终盘点一对一”想要进步的同学
- 高并发场景案例分享（二）count实时查询之坑
- 使用 Three.js 制作一个专属3D奖牌
- 巧用 CSS 实现动态线条 Loading 动画
- Abp vnext EFCore 实现动态上下文 DbSet 踩坑记

 百度智能云

企业级云服务器

305元

立即购买

最新新闻：

- 监管下的百亿市场：密室逃脱陷入焦虑
  - 李开复加持，创新奇智流血上市背后的“红与黑”
  - 奈飞受挫，长视频困局彰显
  - 谷歌对打Meta秘密项目曝光：300人布局元宇宙，专注硬件，进门签署保密协议
  - 疫情这两年，代购怎么活？
- » 更多新闻...

