**什么是 chroot**

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 `/`，即是以根 (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。

**为何使用 chroot**

在经过 chroot 之后，系统读取到的目录和文件将不在是旧系统根下的而是新根下(即被指定的新的位置)的目录结构和文件，因此它带来的好处大致有以下3个：

1. 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。这个一般是在登录 (login) 前使用 chroot，以此达到用户不能访问一些特定的文件。

1. 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。

1. 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。另外，当系统出现一些问题时，我们也可以使用 chroot 来切换到一个临时的系统。

**chroot 的使用**

为了更好的理解 chroot 发挥的作用，我们将尝试指定一个特定的位置进行根目录切换。但是由于在经过 chroot 之后，系统读取到的 bin/ 等与系统相关目录将不再是旧系统根目录下的，而是切换后新根下的目录结构和文件，因此我们有必要准备一些目录结构以及必要的文件。

1. 建立新根目录环境：假定指定新根目录为/home/username/root

#cd /home/username/root

#cp /bin/ . -r

#mk usr

#cp /usr/bin usr -r

#cp /usr/lib/ usr -r

1. 使用chroot

#cd /home/username/root

#chroot .

#pwd

/

1. 编写程序使该进程change the root, 与原文件系统隔离

#define SAFE\_DIR "/home/cr7/root"

int main(void)

{

int ret;

register pid\_t pid;

char \*safedir=NULL;

struct stat sdir\_stat;

register pid\_t waited;

int status;

char \*rets;

char path[100];

if (!stat(SAFE\_DIR, &sdir\_stat) && S\_ISDIR(sdir\_stat.st\_mode))

{

safedir = SAFE\_DIR;

printf("safe\_dir ok\n");

}

else

{

fprintf(stderr, "Helper: %s does not exist ?!\n", SAFE\_DIR);

return -1;

}

pid = syscall(SYS\_clone, CLONE\_FS | SIGCHLD, 0, 0, 0);

switch (pid)

{

// struct rlimit nf;

case -1:

perror("clone");

return -1;

/\* child \*/

case 0:

/\* FIXME: change directory + check permissions first. \*/

rets = getcwd(path, 100);

if(rets)

printf("before chroot path:%s\n", path);

ret = chroot(safedir);

if (ret)

{

perror("Helper: chroot");

exit(EXIT\_FAILURE);

}

ret = chdir("/");

if (ret)

{

perror("Helper: chdir");

exit(EXIT\_FAILURE);

}

printf("Helper: I chrooted you\n");

exit(EXIT\_SUCCESS);

default:

waited = waitpid(pid, &status, 0);

if (waited != pid)

{

perror("waitpid");

exit(EXIT\_FAILURE);

}

else

{

rets = chdir("/");

if (rets)

{

perror("child: chdir");

exit(EXIT\_FAILURE);

}

printf("child exit ok!\n");

rets = getcwd(path, 100);

if(rets)

printf("in main after chroot path:%s\n", path);

printf("here ok, run shell\n");

fprintf(stderr, "Executing sh\n");

execv("bin/sh", NULL);

perror("exec failed");

exit(EXIT\_SUCCESS);

}

}

}

该程序通过子父进程共享FS，在父进程中chroot到指定目录，然后在子进程中运行shell进行验证。