**xinet.d下配置文件的常用参数**

disabl：定义是否启动{no|yes}no表示开启，yes表示关闭

socket\_type {stream|dgram}表示套接字格式，stream是TCP,dgram为udp

protocol：协议类型，这些是需是/etc/protocol里可用的

wait 允许并发数，{yes|no}yes单线程，no多线程

user  运行身份

server 有哪个文件/程序启用该服务

server\_args  传递的参数

only-from 白名单

no-access 黑名单

access-time 定义访问时间

log-type {SYSLOG|FILE} syslog定义日志类型，级别；file定义日志存放位置

bind 监听ip

log-on-success 记录登陆成功的信息

log-on-failure  记录登陆失败的信息

per-source    资源限制，限制某一ip同时连接请求数

phier-source = UNLIMITED表示不做限制

cps = n m 定义每秒最大连接数为n，超出后等待m秒后再尝试连接

**Linux下TFTP安装和使用**

**TFTP协议简介**  
TFTP是用来下载远程文件的最简单网络协议，它其于UDP协议而实现。嵌入式linux的tftp开发环境包括两个方面：一是linux服务器端的tftp-server支持，二是嵌入式目标系统的tftp-client支持。因为u-boot本身内置支持tftp-client，所以嵌入式目标系统端就不用配置了。下面就详细介绍一下linux服务器端tftp-server的配置。

[**1、安装tftp服务器**需要安装xinetd、tftp和tftp-server 3个软件  
1）如果能上网，通过yum安装：  
sudo yum install xinetd  
sudo yum install tftp  
sudo yum install tftp-server  
2）如果不能上网，可以直接安装提供的rpm包：  
sudo rpm -ivh xinetd-<?xml:namespace prefix = st1 />2.3.14-18.fc9.i386.rpm  
sudo rpm -ivh tftp-0.48-3.fc9.i386.rpm  
sudo rpm -ivh tftp-server-0.48-3.fc9.i386.rpm](http://www.embedexpert.com/article/linuxkernel/2010/1020/tftpserverset.html1)

**2、配置**[**tftp服务器**](http://www.embedexpert.com/)  
修改/etc/xinetd.d/tftp文件，将其中的disable=yes改为disable=no。

设置TFTP服务器的根目录，并开启服务。配置文件如下：  
service tftp  
{      socket\_type            = dgram  
        protocol                  = udp  
        wait = no no表示为多线程服务，yes表示为单线程，用于并发连接请求  
        user = root 以root的身份运行此服务  
        server                     = /usr/sbin/in.tftpd 定义有哪个进程起启动此服务，并接受用户访问  
        server\_args           = -s /var/lib/tftpboot/ 根目录  
        disable                   = no  
        per\_source           = 11  
        cps                         = 100 2 表示每秒最多可以有100个用户连接次服务，超出着等待2s重连

access\_times = 08:00-19:30 指定可以使用此服务的时间

only\_from = 192.168.0.0/25 表示仅允许192.168.0.0/25网段的计算机连接次服务  
       flags                       = IPv4  
}

说明：修改项server\_args= -s    <path>    -c，其中<path>处可以改为你的tftp-server的根目录，参数-s指定chroot，-c指定了可以创建文件。

**3、启动tftp服务器并关闭防火墙**/etc/init.d/iptables stop        //关闭防火墙

开启服务：

chkconfig tftp on; service xinetd restart; chkconfig xinetd on;

systemctl restart tftp

**4、查看tftp服务是否开启**  
netstat -a | grep tftp  
显示结果为  
udp 0 0 \*:tftp \*:\*  
表明服务已经开启，就表明tftp配置成功了。

**5、tftp使用**

复制一个文件到tftp服务器目录，然后在主机启动tftp软件，进行简单测试。  
tftp 192.168.1.2  
tftp>get <download file>

tftp>put <upload file>  
tftp>q

**6、tftp命令用法如下**tftp     your-ip-address

【进入TFTP操作】  
connect：连接到远程tftp服务器  
mode：文件传输模式  
put：上传文件  
get：下载文件  
quit：退出  
verbose：显示详细的处理信息  
tarce：显示包路径  
status：显示当前状态信息  
binary：二进制传输模式  
ascii：ascii传送模式  
rexmt：设置包传输的超时时间  
timeout：设置重传的超时时间  
help：帮助信息  
?：帮助信息

**7、如果老是出现“AVC Denial, click icon to view”的错误**，并不能传输文件，需要作如下修改  
修改/etc/sysconfig/selinux,将SELINUX设定为disable  
使用命令setenforce 0让selinux配置文件生效

**Linux中telnet客户端与服务端的安装与配置**

Telnet服务的配置步骤如下:

一、安装telnet软件包(通常要两个）。

　　1、 telnet-client (或 telnet)，这个软件包提供的是 telnet 客户端程序；

　　2、是 telnet-server 软件包，这个才是真正的 Telnet server 软件包！

　 安装之前先检测是否这些软件包已安装，方法如下：

    [root@localhost ~]# rpm -q telnet

    telnet-0.17-38.el5

    telnet客户端一般默认情况是已安装

    [root@localhost ~]# rpm -q telnet-server

    无表示未安装

二、安装telnet-client (或 telnet)

yum安装telnet-server，命令：yum -y install telnet，就能安装了。

三、安装telnet-server

yum方式是最为便捷的在线包安装工具。

安装方式都可以直接搜索关键字，类似于：

命令：yum telnet

[root@crxjtest xinetd.d]# yum list |grep telnet

...

Trying other mirror.

telnet.x86\_64 1:0.17-47.el6 @anaconda-RedHatEnterpriseLinux-201111171049.x86\_64/6.2

telnet-server.x86\_64 1:0.17-47.el6 **local**

发现有telnet-server.x86\_64这个源

[root@crxjtest xinetd.d]# yum install telnet-server.x86\_64

...

Trying other mirror.

Setting up Install Process

Resolving Dependencies

--> Running transaction check

---> Package telnet-server.x86\_64 1:0.17-47.el6 will be installed

--> Processing Dependency: xinetd **for** package: 1:telnet-server-0.17-47.el6.x86\_64

--> Running transaction check

---> Package xinetd.x86\_64 2:2.3.14-33.el6 will be installed

--> Finished Dependency Resolution

Dependencies Resolved

===============================================================================================================

Package Arch Version Repository Size

===============================================================================================================

Installing:

telnet-server x86\_64 1:0.17-47.el6 local 37 k

Installing **for** dependencies:

xinetd x86\_64 2:2.3.14-33.el6 local 120 k

Transaction Summary

===============================================================================================================

Install 2 Package(s)

Total download size: 157 k

Installed size: 312 k

Is this ok [y/N]: y

Downloading Packages:

Setting up **and** reading Presto delta metadata

Processing delta metadata

Package(s) data still to download: 157 k

---------------------------------------------------------------------------------------------------------------

Total 1.2 MB/s | 157 kB 00:00

Running rpm\_check\_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : 2:xinetd-2.3.14-33.el6.x86\_64 1/2

Installing : 1:telnet-server-0.17-47.el6.x86\_64 2/2

Installed products updated.

Installed:

telnet-server.x86\_64 1:0.17-47.el6

Dependency Installed:

xinetd.x86\_64 2:2.3.14-33.el6

Complete!

[root@crxjtest xinetd.d]#

安装成功。

可再搜索遍看是否已经写入环境中。

[root@crxjtest xinetd.d]# rpm -qa | grep telnet

telnet-0.17-47.el6.x86\_64

telnet-server-0.17-47.el6.x86\_64

发现telnet-server已经安装成功。

 [更改配置文件，将telnet服务设置为默认启动，非必须]

查询xinetd.d所管辖的所有配置文件所在目录。

[root@crxjtest xinetd.d]# cd /etc/xinetd.d

[root@crxjtest xinetd.d]# ls

chargen-dgram cvs daytime-stream discard-stream echo-stream tcpmux-server **time**-dgram

chargen-stream daytime-dgram discard-dgram echo-dgram rsync telnet **time**-stream

[root@crxjtest xinetd.d]# vi telnet

# default: on

# description: The telnet server serves telnet sessions; it uses \

# unencrypted username/password pairs for authentication.

service telnet

{

disable = **no**

flags = REUSE

socket\_type = stream

**wait** = **no**

user = root

server = /usr/sbin/in.telnetd

log\_on\_failure += USERID

}

备份telnet文件，再vi打开，将disable值赋为no。

修改完之后，:wq保存并退出

开启telnet服务

[root@crxjtest xinetd.d]# **service** **xinetd** **restart**

**Stopping** **xinetd**: [FAILED]

**Starting** **xinetd**: [ OK ]

可直接start，这样就不会有关闭服务时候报的那个错，毕竟那个时候服务还不存在。

测试telnet服务

[root@crxjtest xinetd.d]# telnet localhost

Trying ::1...

Connected **to** localhost.

Escape character **is** '^]'.

Red Hat Enterprise Linux Server release 6.2 (Santiago)

Kernel 2.6.32-220.el6.x86\_64 on an x86\_64

login:

测试正常。

**chkconfig使用说明**

chkconfig是管理系统服务(service)的命令行工具。所谓系统服务(service)，就是随系统启动而启动，随系统关闭而关闭的程序。

chkconfig可以更新(启动或停止)和查询系统服务(service)运行级信息。更简单一点，chkconfig是一个用于维护/etc/rc[0-6].d目录的命令行工具。

chkconfig 提供5个功能：

1. 设置service启动信息

# chkconfig name on/off/reset

on、off、reset用于改变service的启动信息。   
on表示开启，off表示关闭，reset表示重置。   
默认情况下，on和off开关只对运行级2，3，4，5有效，reset可以对所有运行级有效。 例如，

# chkconfig httpd on

2. 设置service运行级别

# chkconfig --level levels

例如，

# chkconfig --level 2345 httpd on

指定运行级为2,3,4,5   
等级0表示：表示关机   
等级1表示：单用户模式   
等级2表示：无网络连接的多用户命令行模式   
等级3表示：有网络连接的多用户命令行模式   
等级4表示：不可用   
等级5表示：带图形界面的多用户模式   
等级6表示：重新启动

3. 添加service

# chkconfig --add name

添加一个chkconfig管理的service，并在/etc/rc[0-6].d 目录下添加相应的符号链接(symbolic links)。

4. 移除service

# chkconfig --del name

从chkconfig 管理名单中删除该service，并且删除 /etc/rc[0-6].d 目录下所有与之关联的符号链接(symbolic links)。

5. 列出service的启动信息

# chkconfig --list [name]

如果不指定name，会列出所有services的信息。

每个service每个运行级别都会有一个启动和停止脚本；当切换运行级别时，init不会重启已经启动的service，也不会重新停止已经停止的service。

下面举例说明

(1).列出所有服务的启动情况

1. $ chkconfig --list
2. auditd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
3. redis           0:off   1:off   2:off   3:off   4:off   5:off   6:off
4. restorecond     0:off   1:off   2:off   3:off   4:off   5:off   6:off
5. rpcbind         0:off   1:off   2:on    3:on    4:on    5:on    6:off
6. rpcgssd         0:off   1:off   2:off   3:on    4:on    5:on    6:off
7. rpcsvcgssd      0:off   1:off   2:off   3:off   4:off   5:off   6:off
8. rsyslog         0:off   1:off   2:on    3:on    4:on    5:on    6:off
9. saslauthd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
10. smb             0:off   1:off   2:off   3:on    4:off   5:on    6:off
11. 。。。
12. xinetd based services:
13. rsync:          off
14. swat:           off

(2)增加mysqld服务

$ chkconfig --add mysqld

(3)删除mysqld服务

$ chkconfig --del mysqld

(4)设置mysqld运行级别为2,3,4,5

$ chkconfig --level 2345 httpd on

(5)列出mysqld 服务启动信息情况

1. $ chkconfig --list mysqld
2. mysqld          0:off   1:off   2:on    3:on    4:on    5:on    6:off

(6)设置启动信息

$ chkconfig mysqld on

默认的运行级别为2,3,4,5   
实际上，与4中命令作用是一样的。

====================================================

**chkconfig添加服务**

1. 开发一个程序  
首先开发一个软件，使其成为service。代码如下:

1. //capsule.c
2. #include <unistd.h>
3. #include <time.h>
4. #include <sys/types.h>
5. #include <sys/stat.h>
6. #include <fcntl.h>
7. #include <signal.h>
9. int init\_daemon();
11. int main()
12. {
13. init\_daemon();
15. while(1)
16. {
17. sleep(2);
18. }
20. return 0;
21. }

24. int init\_daemon()
25. {
26. int i;
27. pid\_t   pid;
28. if ( (pid = fork()) < 0)
29. {
30. return (-1);
31. }
32. else if (pid)
33. {
34. \_exit(0);                       /\* parent terminates \*/
35. }
36. /\* child 1 continues... \*/
37. if (setsid() < 0)                       /\* become session leader \*/
38. {
39. return (-1);
40. }
41. signal(SIGHUP, SIG\_IGN);
43. if ( (pid = fork()) < 0)
44. {
45. return (-1);
46. }
47. else if (pid)
48. {
49. \_exit(0);                       /\* child 1 terminates \*/
50. }
51. chdir("/");                             /\* change working directory \*/
53. /\* close off file descriptors \*/
54. for (i = 0; i < 255; i++)
55. {
56. close(i);
57. }
58. /\* redirect stdin, stdout, and stderr to /dev/null \*/
59. open("/dev/null", O\_RDONLY);
60. open("/dev/null", O\_RDWR);
61. open("/dev/null", O\_RDWR);
63. return (0);
64. }

init\_daemon()完成进程的精灵化过程，包括脱离终端转入后台等。

编译

# gcc capsule.c –o capsule

将可执行文件拷贝到/usr/local/sbin目录下

# cp capsule /usr/local/sbin/

2.编写service脚本  
service一般通过chkconfig工具进行管理。chkconfig管理的每个service需要在其init.d脚本中添加两行或更多注释行。   
第一行告诉chkconfig 该service运行的默认级别，以及启动和停止的优先权。如果该service不在任何运行级启动，可以设置为”-“。   
第二行包含描述信息。

针对刚才的可执行文件，编写service脚本capsuled如下：

1. #!/bin/bash
2. #
3. # capsuled    A test service program
4. #
5. # chkconfig: - 92 12
6. # description: A test service prog
7. #
8. # @name: capsuled
9. # @author: lanyang
10. # @created: 2017.01.23
11. #
12. # Source function library.
13. . /etc/init.d/functions

16. PROG=capsuled
17. RETVAL=0
18. FULL\_PATH=/usr/local/sbin/capsule
20. start()
21. {
22. echo -n $"Starting $PROG ..."
23. daemon $FULL\_PATH
24. RETVAL=$?
25. echo
26. }
28. stop()
29. {
30. echo -n $"Stopping $PROG ..."
31. killproc $FULL\_PATH
32. RETVAL=$?
33. echo
34. }
36. case "$1" in
37. start)
38. start
39. ;;
40. stop)
41. stop
42. ;;
43. restart)
44. stop
45. sleep 5
46. start
47. RETVAL=$?
48. ;;
49. status)
50. status $FULL\_PATH
51. RETVAL=$?
52. ;;
53. \*)
54. echo $“Usage: $0 {start|stop|restart|status}”
55. exit 1
57. esac
59. exit $RETVAL

3.在/etc/init.d/目录下添加service脚本

1. # cp capsuled /etc/init.d
2. # chmod a+x capsuled

4.添加service，使chkconfig命令可以管理该service

# chkconfig --add capsuled

5.设置service启动运行级别

# chkconfig --level 2345 capsuled on

查看启动信息

1. # chkconfig --list capsuled
2. capsuled        0:off   1:off   2:on    3:on    4:on    5:on    6:off

6.启动service

1. # service capsuled start
2. Starting capsuled ...                                      [  OK  ]
3. # ps -ef | grep capsule
4. root      17729      1  0 18:31 ?        00:00:00 /usr/local/sbin/capsule

7. # service capsuled status
8. capsule (pid 17729) is running...
10. # service capsuled stop
11. Stopping capsuled ...                                      [  OK  ]

7.小结

# service  capsuled start

与

# /etc/init.d/capsuled start

作用是一样的。实际上，前者是通过调用后者实现的。

管理service使用的chkconfig是一个用于维护/etc/rc[0-6].d目录的命令行工具。其中，[0-6]指的是系统的7个运行级别。   
/etc/rc[0-6].d目录，内容全部是链接(symbolic links)，一般链接到/etc/init.d/目录下的某个service脚本文件。   
例如，

1. $ ll /etc/rc5.d/S85httpd
2. lrwxrwxrwx. 1 root root 15 Jul 30  2015 /etc/rc5.d/S85httpd -> ../init.d/httpd

其中，85是启动优先级；

1. $ ll /etc/rc6.d/K15httpd
2. lrwxrwxrwx. 1 root root 15 Jul 30  2015 /etc/rc6.d/K15httpd -> ../init.d/httpd

其中，15是停止优先级。   
类似的，

1. # ll /etc/rc4.d/ | grep capsuled
2. lrwxrwxrwx. 1 root root 18 Jan 23 17:53 S92capsuled -> ../init.d/capsuled
4. # ll /etc/rc5.d/ | grep capsuled
5. lrwxrwxrwx. 1 root root 18 Jan 23 17:53 S92capsuled -> ../init.d/capsuled

92是启动优先级

1. # ll /etc/rc6.d/ | grep capsuled
2. lrwxrwxrwx. 1 root root 18 Jan 23 17:52 K12capsuled -> ../init.d/capsuled

12是停止优先级