

介绍守护进程XINETD与SOCKET程序的含义 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/237/2021_2022__E4_BB_8B_E7_BB_8D_E5_AE_88_E6_c103_237529.htm 1.Xinetd提供被动式的超级服务 也就是服务程序是被使用端所启动，平时则无须存在。例如，ftp, telnetd, pop3,imap, auth等等，这些服务没有人使用时，无须启动。此外，xinetd将socket转换成stdin/stdout，因而使得网路服务程式设计大大简化，您可以只用printf及fgets便可完成处理很复杂的网路协定。 2.一个简单的服务程序

```
goodie #include#include#includechar *cmds[]={ "help", "say", "hello", "bye", "exit", NULL }. int getcmd(char *cmd) { int n=0. while (cmds[n]!=NULL) { if (strncasecmp(cmd,cmds[n],strlen(cmds[n]))==0) return n. n . } return -1. } void main(void) { char buf[1024]. int ok. printf("Welcome to goodie service! "). fflush(stdout). ok=0. do { while (fgets(buf,1023,stdin)==NULL). switch (getcmd(buf)) { case -1: printf("Unknown command! "). break. case 0: printf("How may I help you, sir? "). int j=0. while(cmds[j ]!=NULL)printf( " %s\t\t " ,cmds[j-1]). break. case 1: printf("I will say %s",&.buf[3]). break. case 2: printf("Howre you doing today? "). break. case 3: printf("Si ya, mate! "). ok=1. break. case 4: printf("Go ahead! "). ok=1. break. } fflush(stdout). } while (!ok). }3.配置文件
```

在/etc/services 中加入如下项 goodie 12345/tcp 服务名是 goodie 服务端口是 12345 服务类型是 TCP 在/etc/xinetd.d目录下新检服务文件 goodie 输入内容： service goodie{ socket_type=stream protocol =tcp wait =no server = /{goodie dir}/goodir disable =no #

初始化为开启服务监听}4.启动服务 先停止服务 killall xinetd 启动服务 /usr/sbin/xinetd -f /etc/xinetd.conf 5.开启终端连接 telnet localhost 12345 6.xinetd 的工作原理 通过配置xinetd，然后查看相应的套接字和进程，可以看出，xinetd是这样工作的（针对tcp服务）

A.启动时读取/etc/xinetd.conf文件并为文件中指定的所有服务创建相应的套接字（流或数据报），xinetd能处理的服务的数目依赖于所创建的套接字数目。每个新创建的套接字都被加入到0select调用所用到的描述符集中。

B.对每一个套接字调用bind，绑定服务端口（/etc/services中定义），端口号通过调用getservbyname获得。

C.所有套接字建立后，调用0select等待它们变为可读，当tcp套接字上有数据报到来时变为可读。xinetd在大部分时间阻塞在0select的调用处；

D.xinetd守护进程fork，由子进程处理服务请求；子进程关闭除了要处理的套接字之外的所有描述符，子进程三次调用dup2，把套接字描述符复制到0、1、2，然后关闭原套接字；以后程序对套接字操作就是对0、1、2进行操作；子进程exec执行相应的服务器程序，并将配置文件中的参数传递。

E.因为tcp服务器通常设置nowait标记，表示xinetd在该套接字上再次选择之前，必须等待在该套接字上服务的子进程终止。所以，父进程中的fork返回时，把子进程的进程号记录下来，这样，在子进程终止时，父进程可以用waitpid的返回值查知是那一个子进程；父进程用FD_CLR宏关闭0select使用的描述符集中与这个套接字对应的位，以便不对该套接字0select；当子进程终止时，父进程收到一个SIGCHLD信号，父进程的信号处理程序得到终止子进程的进程号，父进程通过打开描述符集中对应的位恢复对该套接字的0select。

100Test 下载

频道开通，各类考试题目直接下载。详细请访问

www.100test.com