<https://blog.csdn.net/taiyang1987912/article/details/45695157>

要写个检测网络接口链接状态的东西，又不喜欢不断的ping别的地址，也不想调用其他命令行工具来做这个，于是在google了n多内容未果之后，搜刮到个检测工具的源代码。  
  
以下代码在fedora 9 / CentOS 5.2下调试通过:)

**[html]** [view plain](https://blog.csdn.net/taiyang1987912/article/details/45695157) [copy](https://blog.csdn.net/taiyang1987912/article/details/45695157)

1. #include **<sys**/types.h**>**
2. #include **<string.h>**
3. #include **<stdlib.h>**
4. #include **<sys**/ioctl.h**>**
5. #include **<stdio.h>**
6. #include **<errno.h>**
7. #include **<net**/if.h**>**
9. struct ethtool\_value
10. {
11. \_\_uint32\_t cmd;
12. \_\_uint32\_t data;
13. };
15. /\*return 1:has cable;   return 0:no cable\*/
16. int detect\_eth\_cable(char \*ifname)
17. {
18. struct ethtool\_value edata;
19. struct ifreq ifr;
20. int fd = -1, err = 0;
22. memset(&ifr, 0, sizeof(ifr));
23. strcpy(ifr.ifr\_name, ifname);
25. fd = socket(AF\_INET, SOCK\_DGRAM, 0);
26. if (fd **<** **0**) {
27. //perror("Cannot get control socket");
28. return -1;
29. }
30. edata.cmd = 0x0000000A;
31. ifr.ifr\_data = (caddr\_t)&edata;
32. err = ioctl(fd, 0x8946, &ifr);
33. if (err == 0) {
34. fprintf(stdout, "Link detected: %s\n", edata.data ? "yes":"no");
35. } else if (errno != EOPNOTSUPP) {
36. perror("Cannot get link status");
37. }
38. return(edata.data==1 ? 1:0);
39. }
41. int main(int argc, char\*\*argv)
42. {
43. detect\_eth\_cable("p1p1");
44. return 0;
45. }

其他代码：

**[html]** [view plain](https://blog.csdn.net/taiyang1987912/article/details/45695157) [copy](https://blog.csdn.net/taiyang1987912/article/details/45695157)

1. int get\_netportstatus(const char \*interface) {
2. char cmd[1024];
3. char \*tt;
4. FILE \*fp;
5. int devflag;
6. devflag=get\_netflag(interface);
7. if (devflag==DEV\_DOWN) {
8. sprintf(cmd,"ifconfig %s up",interface);
9. system(cmd);
10. }
11. sprintf(cmd,"ethtool %s | grep \"Link detected\" **>** /tmp/eth.temp",interface);
12. system(cmd);
13. if (devflag==DEV\_DOWN) {
14. sprintf(cmd,"ifconfig %s down",interface);
15. system(cmd);
16. }
17. fp=fopen("/tmp/eth.temp","r");
18. if (fp==NULL) {
19. system("rm -rf /tmp/eth.temp");
20. return -1;
21. }
22. fgets(cmd,1024,fp);
23. fclose(fp);
24. system("rm -rf /tmp/eth.temp");
25. tt=strstr(cmd,"no");
26. if (tt!=NULL) return LINK\_DOWN;
27. tt=strstr(cmd,"yes");
28. if (tt!=NULL) return LINK\_UP;
29. return -1;
30. }

**[html]** [view plain](https://blog.csdn.net/taiyang1987912/article/details/45695157) [copy](https://blog.csdn.net/taiyang1987912/article/details/45695157)

1. #include **<sys**/types.h**>**
2. #include **<string.h>**
3. #include **<stdlib.h>**
4. #include **<sys**/ioctl.h**>**
5. #include **<stdio.h>**
6. #include **<errno.h>**
7. #include **<net**/if.h**>**

10. struct ethtool\_value {
11. \_\_uint32\_t      cmd;
12. \_\_uint32\_t      data;
13. };

16. int main(int , char\* [])
17. {
18. struct ethtool\_value edata;
19. int fd = -1, err = 0;
20. struct ifreq ifr;

23. memset(&ifr, 0, sizeof(ifr));
24. strcpy(ifr.ifr\_name, "eth0");
25. fd = socket(AF\_INET, SOCK\_DGRAM, 0);
26. if (fd **<** **0**) {
27. perror("Cannot get control socket");
28. return 70;
29. }

32. edata.cmd = 0x0000000a;
33. ifr.ifr\_data = (caddr\_t)&edata;
34. err = ioctl(fd, 0x8946, &ifr);
35. if (err == 0) {
36. fprintf(stdout, "Link detected: %s\n",
37. edata.data ? "yes":"no");
38. } else if (errno != EOPNOTSUPP) {
39. perror("Cannot get link status");
40. }
41. return 0;
42. }

**[html]** [view plain](https://blog.csdn.net/taiyang1987912/article/details/45695157) [copy](https://blog.csdn.net/taiyang1987912/article/details/45695157)

1. #include **<net** if.h=""**>**  // IFF\_RUNNING
3. //如果网卡已脸上网线，返回0，否则返回-1.
4. int check\_nic(char \*nic)
5. {
6. struct ifreq ifr;
7. int skfd = socket(AF\_INET, SOCK\_DGRAM, 0);
9. strcpy(ifr.ifr\_name, nic\_name);
10. if (ioctl(skfd, SIOCGIFFLAGS, &ifr) **<** **0**)
11. {
12. return -1;
13. }
14. if(ifr.ifr\_flags & IFF\_RUNNING)
15. return 0;  // 网卡已插上网线
16. else return -1;
17. }
18. **</net>**