<http://ju.outofmemory.cn/entry/16484>

在[Linux](http://www.cpplive.com/html/tag/linux" \o "查看 Linux 中的全部文章" \t "_blank)下开发网络程序时，经常会遇到需要取本地网络接口名、[IP](http://www.cpplive.com/html/tag/ip" \o "查看 IP 中的全部文章" \t "_blank)、[广播](http://www.cpplive.com/html/tag/broadcast" \o "查看 广播 中的全部文章" \t "_blank)[地址](http://www.cpplive.com/html/tag/address)、子网掩码或者MAC地址等信息的需求，最常见的办法是配合宏SIOCGIFHWADDR、SIOCGIFADDR、SIOCGIFBRDADDR与SIOCGIFNETMASK作为参数调用[函数](http://www.cpplive.com/html/tag/function" \o "查看 函数 中的全部文章" \t "_blank)ioctl分别获得MAC地址、IP地址、广播地址与子网掩码来实现。一次性获取此类信息的[C语言](http://www.cpplive.com/html/tag/c-language" \o "查看 C语言 中的全部文章" \t "_blank)代码实现如下。

#include <stdio.h>

#include <string.h>

#include <net/if.h>

#include <sys/ioctl.h>

#include <arpa/inet.h>

#include <errno.h>

int getLocalInfo(void)

{

int fd;

int interfaceNum = 0;

struct ifreq buf[16];

struct ifconf ifc;

struct ifreq ifrcopy;

char mac[16] = {0};

char ip[32] = {0};

char broadAddr[32] = {0};

char subnetMask[32] = {0};

if ((fd = socket(AF\_INET, SOCK\_DGRAM, 0)) < 0)

{

perror("socket");

close(fd);

return -1;

}

ifc.ifc\_len = sizeof(buf);

ifc.ifc\_buf = (caddr\_t)buf;

if (!ioctl(fd, SIOCGIFCONF, (char \*)&ifc))

{

interfaceNum = ifc.ifc\_len / sizeof(struct ifreq);

printf("interface num = %d\n", interfaceNum);

while (interfaceNum-- > 0)

{

printf("\ndevice name: %s\n", buf[interfaceNum].ifr\_name);

//ignore the interface that not up or not runing

ifrcopy = buf[interfaceNum];

if (ioctl(fd, SIOCGIFFLAGS, &ifrcopy))

{

printf("ioctl: %s [%s:%d]\n", strerror(errno), \_\_FILE\_\_, \_\_LINE\_\_);

close(fd);

return -1;

}

//get the mac of this interface

if (!ioctl(fd, SIOCGIFHWADDR, (char \*)(&buf[interfaceNum])))

{

memset(mac, 0, sizeof(mac));

snprintf(mac, sizeof(mac), "%02x%02x%02x%02x%02x%02x",

(unsigned char)buf[interfaceNum].ifr\_hwaddr.sa\_data[0],

(unsigned char)buf[interfaceNum].ifr\_hwaddr.sa\_data[1],

(unsigned char)buf[interfaceNum].ifr\_hwaddr.sa\_data[2],

(unsigned char)buf[interfaceNum].ifr\_hwaddr.sa\_data[3],

(unsigned char)buf[interfaceNum].ifr\_hwaddr.sa\_data[4],

(unsigned char)buf[interfaceNum].ifr\_hwaddr.sa\_data[5]);

printf("device mac: %s\n", mac);

}

else

{

printf("ioctl: %s [%s:%d]\n", strerror(errno), \_\_FILE\_\_, \_\_LINE\_\_);

close(fd);

return -1;

}

//get the IP of this interface

if (!ioctl(fd, SIOCGIFADDR, (char \*)&buf[interfaceNum]))

{

snprintf(ip, sizeof(ip), "%s",

(char \*)inet\_ntoa(((struct sockaddr\_in \*)&(buf[interfaceNum].ifr\_addr))->sin\_addr));

printf("device ip: %s\n", ip);

}

else

{

printf("ioctl: %s [%s:%d]\n", strerror(errno), \_\_FILE\_\_, \_\_LINE\_\_);

close(fd);

return -1;

}

//get the broad address of this interface

if (!ioctl(fd, SIOCGIFBRDADDR, &buf[interfaceNum]))

{

snprintf(broadAddr, sizeof(broadAddr), "%s",

(char \*)inet\_ntoa(((struct sockaddr\_in \*)&(buf[interfaceNum].ifr\_broadaddr))->sin\_addr));

printf("device broadAddr: %s\n", broadAddr);

}

else

{

printf("ioctl: %s [%s:%d]\n", strerror(errno), \_\_FILE\_\_, \_\_LINE\_\_);

close(fd);

return -1;

}

//get the subnet mask of this interface

if (!ioctl(fd, SIOCGIFNETMASK, &buf[interfaceNum]))

{

snprintf(subnetMask, sizeof(subnetMask), "%s",

(char \*)inet\_ntoa(((struct sockaddr\_in \*)&(buf[interfaceNum].ifr\_netmask))->sin\_addr));

printf("device subnetMask: %s\n", subnetMask);

}

else

{

printf("ioctl: %s [%s:%d]\n", strerror(errno), \_\_FILE\_\_, \_\_LINE\_\_);

close(fd);

return -1;

}

}

}

else

{

printf("ioctl: %s [%s:%d]\n", strerror(errno), \_\_FILE\_\_, \_\_LINE\_\_);

close(fd);

return -1;

}

close(fd);

return 0;

}

int main(void)

{

getLocalInfo();

return 0;

}

使用ioctl函数虽然可以获取所有的信息，但是使用起来比较麻烦，如果不需要获取MAC地址，那么使用getifaddrs函数来获取更加方便与简洁。值得一提的是，在MacOS或[iOS](http://www.cpplive.com/html/tag/ios" \o "查看 iOS 中的全部文章" \t "_blank)系统上（如[iPhone](http://www.cpplive.com/html/tag/iphone" \o "查看 iPhone 中的全部文章" \t "_blank)程序开发），上述iotcl函数没法获得mac地址跟子网掩码，这个使用，使用getifaddrs函数便更有优势了。下面是使用getiaddrs函数获取网卡信息的C语言代码实现。

#include <stdio.h>

#include <ifaddrs.h>

#include <arpa/inet.h>

int getSubnetMask()

{

struct sockaddr\_in \*sin = NULL;

struct ifaddrs \*ifa = NULL, \*ifList;

if (getifaddrs(&ifList) < 0) return -1;

for (ifa = ifList; ifa != NULL; ifa = ifa->ifa\_next)

{

if(ifa->ifa\_addr->sa\_family == AF\_INET)

{

printf("\n>>> interfaceName: %s\n", ifa->ifa\_name);

sin = (struct sockaddr\_in \*)ifa->ifa\_addr;

printf(">>> ipAddress: %s\n", inet\_ntoa(sin->sin\_addr));

sin = (struct sockaddr\_in \*)ifa->ifa\_dstaddr;

printf(">>> broadcast: %s\n", inet\_ntoa(sin->sin\_addr));

sin = (struct sockaddr\_in \*)ifa->ifa\_netmask;

printf(">>> subnetMask: %s\n", inet\_ntoa(sin->sin\_addr));

}

}

freeifaddrs(ifList);

return 0;

}

int main(void)

{

getSubnetMask();

return 0;

}