

Programowanie usług sieciowych												
05	Temat:	<i>Zarządzanie lokalną konfiguracją siecią</i>					Zadania:				Data:	
	Autor:	Adrian Janus					1	2	3	4	5	2019-03-26
	Autor:	Kamil Janus					e	e	e	e	e	16:15-17:45

Zadanie 1. Parametry interfejsów sieciowych.

Program listuje nam interfejsy. Wspomniana w zadaniu alokacja pamięci jest wymuszona nieznajomością liczby interfejsów w systemie.

Output programu:

```
mint@mint:~/Desktop/pus5/src$ sudo ./iolist
Interface: lo <UP NO-PROMISC>
        address: 127.0.0.1, netmask: 255.0.0.0
Interface: enp0s3 <UP NO-PROMISC>
        address: 10.0.2.15, netmask: 255.255.255.0
```

Program pokazuje nam stosunkowo mało, bo jedynie nazwę, adres, maskę oraz tryb pracy interfejsu (czy jest w trybie nasłuchu tylko swoich adresów czy wszystkich).

W porównaniu ifconfig w podstawowej wersji daje nam jeszcze adres ipv6, sieć, mac, liczbę pakietów otrzymanych i przesłanych i błędy z nimi związane, mtu oraz tryby pracy bardziej szczegółowe. Nie podaje informacji o no promisc.

Ip address nie podaje danych o przepływie pakietów, ale za to podaje długości kolejek oraz zalecany czas życia adresu interfejsu.

Zadanie 2. System plików proc.

Program wypisuje wiadomości o interfejsach ipv6 zawarte w katalogu proc.

Output programu:

```
mint@mint:~/Desktop/pus5/src$ ./proclist
  Name  Index      Address/prefix length      Scope
  lo     1      0000:0000:0000:0000:0000:0000:0001/128      Host
  enp0s3 2      fe80:0000:0000:0000:0a00:27ff:fe37:6fb8/64      Link
Interface list (if_nameindex):
  lo     1
  enp0s3 2
```

Adress scope informuje nas gdzie adres obowiązuje (np. host dla adresu localhost).

Lista nie musi być zgodna. Koniec danych w strukturze jest wskazywany przez indeks zerowy więc dane z pliku mogą być uszkodzone. Oprócz tego część adresów może ulegać zmianą w czasie.

Zadanie 3. Modyfikacja tablicy ARP.

Program zmienia adres mac w tablicy arp.

Przed zmianą:

```
192.168.113.4 > 192.168.113.3: ICMP echo reply, id 1899, seq 4, length 64
17:41:37.960752 08:00:27:35:d2:c9 > 08:00:27:83:c2:b7, ethertype ARP (0x0806), l
length 60: Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.113.4 tell 192
.168.113.3, length 46
```

Jak widać adres mac to 80:00:27:83:c2:b7.

Po Zmianie na 12:12:12:12:12:

```
192.168.113.3 > 192.168.113.4: ICMP echo request, id 2178, seq 3, length 64
17:50:07.402903 08:00:27:35:d2:c9 > 12:12:12:12:12:12, ethertype IPv4 (0x0800),
length 98: (tos 0x0, ttl 64, id 44969, offset 0, flags [DF], proto ICMP (1), len
gth 84)
192.168.113.3 > 192.168.113.4: ICMP echo request, id 2178, seq 4, length 64
17:50:08.427182 08:00:27:35:d2:c9 > 12:12:12:12:12:12, ethertype IPv4 (0x0800),
length 98: (tos 0x0, ttl 64, id 45123, offset 0, flags [DF], proto ICMP (1), len
gth 84)
192.168.113.3 > 192.168.113.4: ICMP echo request, id 2178, seq 5, length 64
17:50:09.452106 08:00:27:35:d2:c9 > 12:12:12:12:12:12, ethertype IPv4 (0x0800),
length 98: (tos 0x0, ttl 64, id 45257, offset 0, flags [DF], proto ICMP (1), len
gth 84)
```

Jak widzimy adres się zmienił, a w snifferze nie ma pakietów icmp reply. Oznacza to, że pingowany nie zaakceptował pakietu pingującego jako swojego przez błędny adres mac.

Zadanie 4. Konfiguracja adresu MAC i MTU.

Ifconfig przed:

```
mint@mint:~/Desktop/pus5/src$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe37:6fb8 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:37:6f:b8 txqueuelen 1000 (Ethernet)
    RX packets 45025 bytes 32721086 (32.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 25024 bytes 2249664 (2.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 70111 bytes 4023571 (4.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 70111 bytes 4023571 (4.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Uzupełniamy pole nazwy interfejsu struktury ifreq:

```
strcpy(ifreqstruct.ifr_ifrn.ifrn_name, argv[1]);
```

Uzupełnienie pola adresu oraz wywołanie ioctl:

```
retval = sscanf(
    argv[2], "%02x:%02x:%02x:%02x:%02x:%02x",
    (unsigned char*)&ifreqstruct1.ifr_ifru.ifru_hwaddr.sa_data[0],
    (unsigned char*)&ifreqstruct1.ifr_ifru.ifru_hwaddr.sa_data[1],
    (unsigned char*)&ifreqstruct1.ifr_ifru.ifru_hwaddr.sa_data[2],
    (unsigned char*)&ifreqstruct1.ifr_ifru.ifru_hwaddr.sa_data[3],
    (unsigned char*)&ifreqstruct1.ifr_ifru.ifru_hwaddr.sa_data[4],
    (unsigned char*)&ifreqstruct1.ifr_ifru.ifru_hwaddr.sa_data[5]
);

if (retval != 6) {
    fprintf(stderr, "Invalid address format!\n");
    exit(EXIT_FAILURE);
}

retval = ioctl(sockfd, SIOCSIFHWADDR, &ifreqstruct1);
if (retval == -1) {
    perror("qioclt()");
    exit(EXIT_FAILURE);
}
```

Zmiana MTU:

```
ifreqstruct1.ifr_ifru.ifru_mtu = atoi(argv[3]);
retval = ioctl(sockfd, SIOCSIFMTU, &ifreqstruct1);
```

Ifconfig po zmianie interfejsu w ten sposób:

```
mint@mint:~/Desktop/pus5/src$ sudo ./iomac enp0s3 34:34:45:79:34:54 1000
stary mac: 08:00:27:37:6f:b8
stare mtu: 1500
zmieniono macnowy mac: 34:34:45:79:34:54
zmieniono mtunowe mtu: 1000
mint@mint:~/Desktop/pus5/src$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1000
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    ether 34:34:45:79:34:54 txqueuelen 1000 (Ethernet)
```

Zadanie 5. Zarządzanie adresami IPv4.

- Dla add:

Sprawdzenie czy to alias, a jeśli tak to czy jest up poprzez szukanie dwukropka, a następnie sprawdzenie, czy jest ustawiona flaga IFF_UP:

```
for(i = 0; i < sizeof(argv[1]); i++)
    if(argv[1][i] == ':') {printf("4\n");break;}
    else tmp[i] = argv[1][i];
if (i != sizeof(argv[1]))
{
    printf("1\n");
    tmp[i] = '\0';
    strcpy(ifreqstruct1.ifr_ifrn.ifrn_name, tmp);
    retval = ioctl(sockfd, SIOCGIFFLAGS, &ifreqstruct1);
    if (retval == -1) {
        perror("lioclt()");
        exit(EXIT_FAILURE);
    }
    if( ifreqstruct1.ifr_ifru.ifru_flags & IFF_UP == 0 ) {printf("interfejs down"); exit(EXIT_FAILURE);}
}
```

Ustawienie adresu:

```
retval = ioctl(sockfd, SIOCSIFADDR, &ifreqstruct);
```

Ustawienie maski:

```
retval = ioctl(sockfd, SIOCSIFNETMASK, &ifreqstruct);
```

- Dla down:

Następuje sprawdzenie czy interfejs ma ustawioną flagę IFF_UP, jeżeli taki interfejs istnieje oraz flaga jest ustawiona to zdejmujemy flagę. Jest to jedyny sposób wyłączenia interfejsu:

```
retval = ioctl(sockfd, SIOCGIFFLAGS, &ifreqstruct1);
if (retval == -1) {
    perror("lioclt()");
    exit(EXIT_FAILURE);
}
if( ifreqstruct1.ifr_ifru.ifru_flags & IFF_UP != 0 )
    ifreqstruct1.ifr_ifru.ifru_flags &= !IFF_UP;
retval = ioctl(sockfd, SIOCSIFFLAGS, &ifreqstruct1);
```

Ip adres po wywołaniu z add:

```

mint@mint:~/Desktop/pus5/src$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:37:6f:b8 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 82847sec preferred_lft 82847sec
    inet 10.0.2.17/24 brd 10.0.2.255 scope global secondary enp0s3:0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe37:6fb8/64 scope link
        valid_lft forever preferred_lft forever

```

Ipaddress po wywołaniu dwon:

```

mint@mint:~/Desktop/pus5/src$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:37:6f:b8 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 82766sec preferred_lft 82766sec
    inet6 fe80::a00:27ff:fe37:6fb8/64 scope link
        valid_lft forever preferred_lft forever

```

Adres drugorzędny się usunął.