

## Grundidee

Meine Grundidee war die Erstellung von der Routingtabelle. Ich wollte ausserdem ein allgemeines GUI erstellen für meine Ausgaben auf der Konsole. Als möglichen Mehrwert wollte ich implementieren, dass die Routingtabelle aktiv bearbeitet werden kann. Im Moment funktioniert das ganze Unterfangen noch nicht.

## RoutingTable Ausgabe

```
#####
#          PROG C          #
#          bachmste        #
#          routing table   #
#####
```

Netadress	Gateway	Port	Netmask
160.85.18.0	-> 0.0.0.0	1	255.255.255.240
160.85.19.0	-> 0.0.0.0	2	255.255.255.0
160.85.16.0	-> 0.0.0.0	0	255.255.254.0
0.0.0.0	-> 160.85.16.1	0	0.0.0.0

## headerFile

Im headerFile sind die diversen Verbindungen definiert. Ausserdem ist die struct für die routingtable definiert. Ausserdem muss die routingTable auch im Moment noch hier angepasst werden.

## headerFile

```
#define IPA(p1, p2, p3, p4) (p1 << 24 | (p2 << 16) | (p3 << 8) | (p4))
#define _IPA(ip) ((ip >> 24) & 0xFF), ((ip >> 16) & 0xFF), ((ip >> 8) & 0xFF), (ip & 0xFF)
#define Default 0
#define Direct 0

typedef unsigned int u32;
typedef unsigned int u8;
typedef struct {
    u32 netAddress;
    u32 netMask;
    u8 port :8;
    u32 gateway;
} routingTable;

routingTable rt[] = {
    {IPA(160, 85, 18, 0), IPA(255, 255, 255, 240), 1, Direct},
    {IPA(160, 85, 19, 0), IPA(255, 255, 255, 0), 2, Direct},
    {IPA(160, 85, 16, 0), IPA(255, 255, 254, 0), 0, Direct},
    {Default, Default, 0, IPA(160, 85, 16, 1)}
};
```

```
printRoutingTable
```

Die Ausgabefunktion kreiert denn zum Schluss noch die Ausgabe, wie oben gezeigt.

```
printRoutingTable
```

```
(void) printf("Netaddress\t Gateway \t\tPort\t\t Netmask\n");  
(void) printf("-----\n");  
for(i=0; i < entryCount; i++) {  
    (void) printf("%u.%u.%u.%u \t-> %u.%u.%u.%u \t\t %u\t %u.%u.%u.%u\n", _IPA(rt[i].  
netAddress),  
_IPA(rt[i].gateway), rt[i].port, _IPA(rt[i].netMask));  
}  
  
(void) printf("-----\n");  
  
return EXIT_SUCCESS;  
}
```