

**Vysoké učení technické v Brně**  
**Fakulta informačních technologií**

**IPK - Počítačové komunikace a sítě**  
**2017/2018**

**Projekt 1.**

**Varianta 1: Klient-server pro získání informace o uživateli**

## Obsah

1.	Popis aplikace .....	3
1.1.	Server.....	3
1.2.	Klient.....	3
2.	Popis aplikačního protokolu .....	3
2.1.	Získávání hodnot z /etc/passwd .....	3
2.2.	Ukončení spojení .....	4
3.	Demonstrace činnosti.....	4

# 1. Popis aplikace

## 1.1. Server

Při implementování serverové části jsem postupoval podle přednášek. Nejprve se zpracují argumenty ve funkci *arguments(...)*. Pak se vytvoří *socket* a spojí se pomocí *bind()* se zadaným portem. Pak následuje *listen()* a *accept()*. Welcome socket pak vytvoří nový socket a ten obslouží požadavek klienta.

## 1.2. Klient

Klient také zpracovává argumenty pomocí funkce *arguments(...)*. Pomocí *getaddrinfo()* se zjistí jestli se jedná o IPv4 nebo o IPv6 adresu. Pokud je ale zadáno doménové jméno, tenhle krok bude neúspěšný a klient se pokusí dostat z doménového jména IP adresu a to pomocí *gethostbyname()*. Z nějakých neznámých důvodů ale tenhle převod nefunguje na serveru eva, na merlinovy to jde...

Pak se pokusí znova zjistit pomocí *getaddrinfo()* jestli se jedná o IPv4 nebo IPv6. Podle toho se pak vytvoří socket (AF\_INET nebo AF\_INET6). Pomocí *connect()* se pokouší připojit na server, který uživatel zadal v argumentech a také na konkrétní port.

Dále se postupuje podle vytvořeného aplikačního protokolu.

# 2. Popis aplikačního protokolu

Hlavní částí aplikačního protokolu je struktura a enum, které se nacházejí v souboru msg.h.

```
typedef enum {TYPE, SEARCH, DATA, END} msg_type;

typedef struct
{
    msg_type type;
    char string[64];
} msg;
```

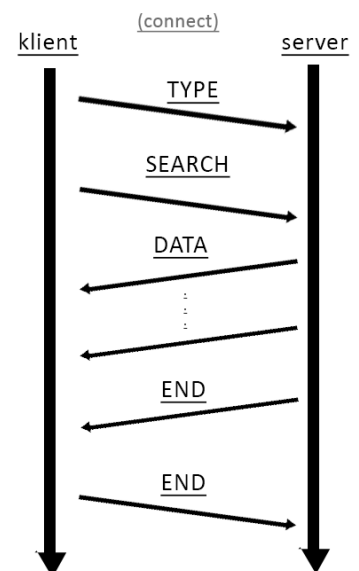
Klient posílá zprávu jako první. Pomocí funkce *send()* se posílá první struktura typu msg. Proměnná type je nastavená na hodnotu TYPE a v poli string se nachází hodnota 'n', 'f' nebo 'l'. První zpráva takhle předá informaci o přepínači, který zadal uživatel a tím pádem o informacích, které má server poslat nazpět.

Klient pak posílá druhou zprávu. V struktuře tentokrát bude type nastaven na hodnotu SEARCH a v poli string bude řetězec, který uživatel hledá, to znamená hodnotu přepínačů n, f nebo l, které jsme získali z argumentů.

Server si tuto informaci uloží pak otevře soubor /etc/passwd.

## 2.1. Získávání hodnot z /etc/passwd

Server používá rozdílné funkce pro získávání dat ze souboru. V případě přepínačů N a F se používá funkce *getNF()*. Soubor se čítá po řádcích pomocí funkce *fscanf()*,



kteřá je uvnitř cyklu a ukládá do příslušných proměnných hodnoty. Pro každý řádek se porovnává hodnota řetězce, který zadal klient s jménem uživatele. Pokud se najde shoda, funkce vrátí podle přepínače N nebo F příslušnou informaci, informaci o uživateli v případě N nebo cestu k adresáři v případě F.

V případě přepínače L se volá funkce *getL()* a také použije funkce *fscanf()* uvnitř cyklu na načítání dat. V každé iteraci se ale volá funkce *checkPrefix()*, která porovná počáteční znaky jména uživatele ze souboru s daným prefixem a vrací *true* nebo *false*. Když dané jméno splňuje tuhle podmínku, hned se pošle pomocí *send()* klientovi.

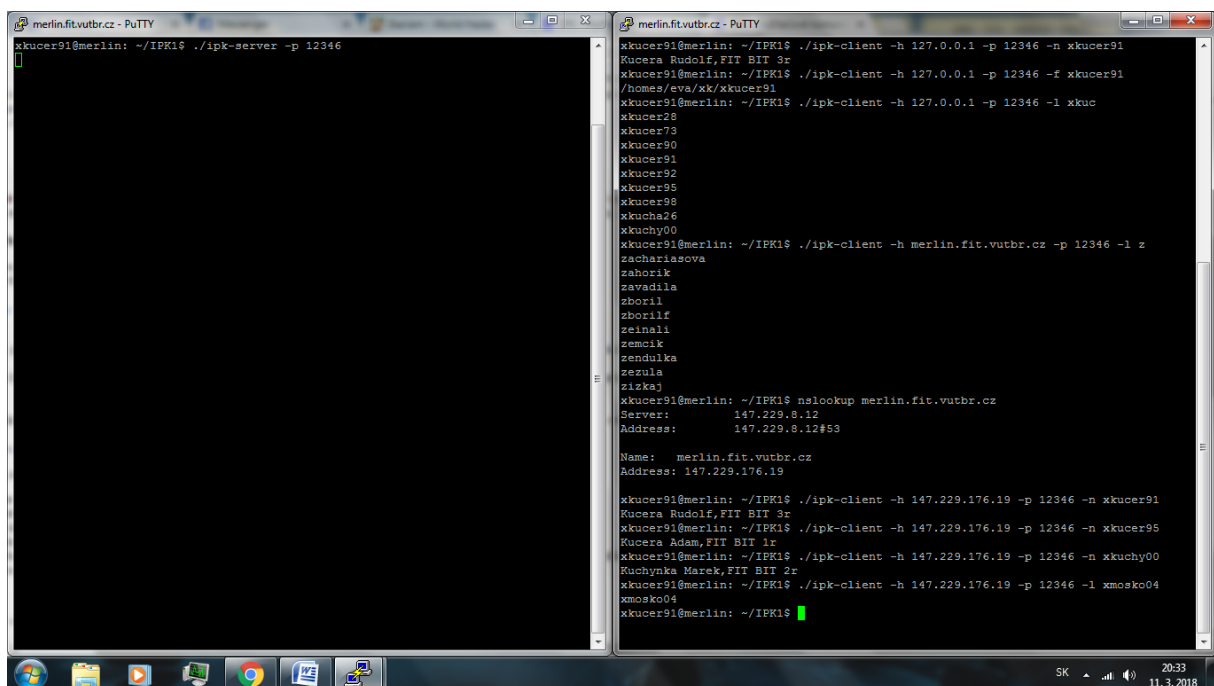
Vytvoří se struktura msg a proměnná type bude nastavena na DATA. Ve proměnné string pak bude toto jméno (pouze jedno). Takže pro každý řádek se vyšle jeden paket.

V případě N a F kdy funkce *getNF()* neposílá zprávy ale jen vrací řetězec, se zpráva pošle až mimo těla této funkce.

## 2.2. Ukončení spojení

Po odeslání dat (struktur s hodnotou type DATA) server pošle zprávu s typem END. Klient pak potvrdí ukončení a to tak že také pošle zprávu typu END. V tomhle případě se ignoruje hodnota v proměnné string.

## 3. Demonstrace činnosti



The image shows two terminal windows side-by-side. The left window is titled 'merlin.fit.vutbr.cz - PuTTY' and shows a server prompt 'xkucer91@merlin: ~/IPK1\$ ./ipk-server -p 12346'. The right window is also titled 'merlin.fit.vutbr.cz - PuTTY' and shows a client prompt 'xkucer91@merlin: ~/IPK1\$ ./ipk-client -h 127.0.0.1 -p 12346 -n xkucer91'. The client window displays a list of names: 'Kucera Rudolf, FIT BIT 3r', 'Kucera Adam, FIT BIT 1r', 'Kuchynka Marek, FIT BIT 2r', and 'xmosko04'. It then shows a series of names: 'xkucha26', 'xkucer90', 'xkucer91', 'xkucer92', 'xkucer95', 'xkucer98', 'xkucha26', 'xkuchy00', 'xkucer91@merlin: ~/IPK1\$ ./ipk-client -h merlin.fit.vutbr.cz -p 12346 -l z', 'zachariasova', 'zahorik', 'zavadila', 'zboril', 'zborilf', 'zeinali', 'zemcik', 'zendulka', 'zezula', 'zizka3'. Below this, it shows a 'nslookup' command and its output: 'Server: 147.229.8.12', 'Address: 147.229.8.12#53', 'Name: merlin.fit.vutbr.cz', 'Address: 147.229.176.19'. Finally, it shows a series of names: 'xkucer91@merlin: ~/IPK1\$ ./ipk-client -h 147.229.176.19 -p 12346 -n xkucer91', 'Kucera Rudolf, FIT BIT 3r', 'xkucer91@merlin: ~/IPK1\$ ./ipk-client -h 147.229.176.19 -p 12346 -n xkucer95', 'Kucera Adam, FIT BIT 1r', 'xkucer91@merlin: ~/IPK1\$ ./ipk-client -h 147.229.176.19 -p 12346 -n xkuchy00', 'Kuchynka Marek, FIT BIT 2r', 'xkucer91@merlin: ~/IPK1\$ ./ipk-client -h 147.229.176.19 -p 12346 -l xmosko04', and 'xmosko04'. The client window ends with 'xkucer91@merlin: ~/IPK1\$'.

Na merlinovi vše funguje tak jak má. Na eve nastávají problémy.

Když se klient z eve pokouší připojit na merlina tak to jde. U klienta na eve ale nejde funkce *gethostbyname()*, nikdy nic nevrací proto klient umí pracovat pouze s IP adresami.

Server na eve nejde vůbec.