

# Varianta 1: Klient-server pro získání informace o uživateli (Ryšavý)

**Společná část popisu:** Vypracujte jednoduchou klient-server aplikaci dle uvedeného zadání.

- Odevzdejte Váš projekt jako archiv s názvem odpovídající Vašemu loginu. Projekty odevzdané po termínu nebudou hodnoceny.
- Odevzdáváte zdrojové soubory projektu
- Projekt musí obsahovat soubor Makefile pro přeložení pomocí make (gmake) ([http://www.linuxsoft.cz/article.php?id\\_article=722](http://www.linuxsoft.cz/article.php?id_article=722))
- Projekt musí být přeložitelný pro uvedený systém (Linux, Unix)
- Součástí projektu je také dokumentace ve formátu plain text, Markdown, nebo PDF.

## Popis varianty: ZADÁNÍ:

Vaším úkolem je:

[1] Seznamte se s kostrami kódů pro programování klientských a serverových síťových aplikací (přednáška třetí) za použití BSD socketů. Navrhněte vlastní aplikační protokol realizující přenos informací o uživateli na straně serveru a relevantní informace k projektu uveďte v dokumentaci. (8 bodů)

[2] Naprogramujte jak klientskou, tak serverovou aplikaci v C/C++ realizující zprostředkování informace o uživateli na serveru. (12 bodů)

## KONVENCE ODEVZDÁVANÉHO ZIP ARCHIVU xlogin00.zip

- dokumentace.pdf - výstupy zadání [1]
- readme - základní informace, případná omezení projektu
- Makefile
- \*.c, \*.cpp, \*.cc, \*.h - zdrojové a hlavičkové soubory výstupů zadání [2]

### Ad [1]

Navrhněte vlastní aplikační protokol, kterými poté spolu budou komunikovat klient a server z bodu [2] tohoto zadání. Tento protokol bude sloužit pro přenos informací o uživateli na serveru. Informace o uživateli bude server získávat ze souboru /etc/passwd.

V dobré dokumentaci se očekává následující: titulní strana, obsah, logické strukturování textu, výcuc relevantních informací z nastudované literatury, popis aplikačního protokolu vhodnou formou, popis zajímavějších pasáží implementace, demonstrace činnosti implementovaných aplikací, normovaná bibliografie.

### Ad [2]

Spuštění klienta předpokládá provedení pouze jedné operace. Komunikace mezi serverem a klientem bude spolehlivá. Server by měl umět obsloužit více požadavků.

Konvence jména klientské aplikace a jejich povinných vstupních parametrů:

```
./ipk-client -h host -p port [-n|-f|-l] login
```

- *host* (IP adresa nebo fully-qualified DNS name) identifikace serveru jakožto koncového bodu komunikace klienta;
- *port* (číslo) cílové číslo portu;
- *-n* značí, že bude vráceno plné jméno uživatele včetně případných dalších informací pro uvedený login (User ID Info);
- *-f* značí, že bude vrácena informace o domácím adresáři uživatele pro uvedený login (Home directory);
- *-l* značí, že bude vrácen seznam všech uživatelů, tento bude vypsán tak, že každé uživatelské jméno bude na zvláštním řádku; v tomto případě je login nepovinný. Je-li však uveden bude použit jako prefix pro výběr uživatelů.
- *login* určuje přihlašovací jméno uživatele pro výše uvedené operace.

např.

```
./ipk-client -h eva.fit.vutbr.cz -p 55555 -n rysavy
```

Rysavy Ondrej,UIFS,541141118

```
./ipk-client -h eva.fit.vutbr.cz -p 55555 -f rysavy
```

/homes/kazi/rysavy

```
./ipk-client -h host -p port -l
```

avahi  
bacova  
barabas

...

zezula  
zizkaj

```
./ipk-client -h host -p port -l xa
```

xabduk00  
xabdul03  
...  
xaygun00

Konvence jména serverové aplikace a jejich povinných vstupních parametrů:

```
./ipk-server -p port
```

- *port* (číslo) číslo portu, na kterém server naslouchá na připojení od klientů.

např.

```
./ipk-server -p 55555
```

## DOPORUČENÍ

- V případě výskytu chyby (neexistující login, problém v komunikaci, chybné parametry), bude ta vypsána na STDERR a na STDOUT nebude tisknut
- Výsledky vaší implementace by měly být co možná nejvíce multiplatformní mezi OS založenými na unixu, ovšem samotné přeložení projektu a ft
- Všechny implementované programy by měly být použitelné a řádně komentované. Pokud už přejímáte zdrojové kódy z různých tutoriálů či příkl
- Pro snadné parsování vstupních argumentů se doporučuje použít funkci `getopt()`.
- Ukončení serverové aplikace z bodu [2] bude vázáno na SIGINT signál (tedy Ctrl+C).
- Projekt bude opravován ručně. Počítejte tedy s nejzazším možným termínem oprav a reklamací určených garantem předmětu.
- Výsledky vaší implementace by měly být co možná nejvíce multiplatformní mezi OS založenými na unixu (dbejte na to zejména při vytváření M
- Do souboru Readme uveďte případná omezení funkcionality vašeho projektu - na dokumentovanou chybu se nahlíží v lepším světle než na nedok
- Snažte se o jednoduchý přehledný kód. Tento projekt nevyžaduje žádné složitosti.

## LITERATURA:

- O. Ryšavý, J. Ráb, *IPK - BSD schránky - 3. přednáška*
- K. Sollins, *The TFTP Protocol (revision 2)*, <https://tools.ietf.org/html/rfc1350>
- J. Postel, J. Reynolds, *FILE TRANSFER PROTOCOL (FTP)*, <https://tools.ietf.org/html/rfc959>
- P. Grygárek, *Softwarová rozhraní systémů UNIX pro přístup k síťovým službám*, <http://www.cs.vsb.cz/grygarek/LAN/sockets.html>
- Understanding /etc/passwd, Online: <https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/>