

Vytvořte komunikující aplikaci podle konkrétní vybrané specifikace pomocí síťové knihovny BSD sockets (pokud není ve variantě zadání uvedeno jinak). Projekt bude vypracován v jazyce C/C++. Pokud individuální zadání nespecifikuje vlastní referenční systém, musí být projekt přeložitelný a spustitelný na serveru merlin.fit.vutbr.cz pod operačním systémem GNU/Linux. Program by měl být přenositelný. Hodnocení projektů může probíhat na jiném počítači s nainstalovaným OS GNU/Linux, včetně jiných architektur než Intel/AMD, distribucí či verzí knihoven. Pokud vyžadujete minimální verzi knihovny (dostupnou na serveru merlin), jasně tuto skutečnost označte v dokumentaci a README.

Vypracovaný projekt uložený v archívu .tar a se jménem xlogin00.tar odevzdejte elektronicky přes IS. Soubor nekomprimujte.

- **Termín odevzdání je 15.11.2021 (hard deadline).** Odevzdání e-mailem po uplynutí termínu, dodatečné opravy či doplnění kódu není možné.
- Odevzdaný projekt musí obsahovat:
  1. soubor se zdrojovým kódem (dodržujte jména souborů uvedená v konkrétním zadání),
  2. funkční *Makefile* pro překlad zdrojového souboru,
  3. dokumentaci (soubor *manual.pdf*), která bude obsahovat uvedení do problematiky, návrhu aplikace, popis implementace, základní informace o programu, návod na použití. V dokumentaci se očekává následující: titulní strana, obsah, logické strukturování textu, přehled nastudovaných informací z literatury, popis zajímavějších pasáží implementace, použití vytvořených programů a literatura.
  4. soubor *README* obsahující krátký textový popis programu s případnými rozšířeními/omezeními, příklad spuštění a seznam odevzdaných souborů,
  5. další požadované soubory podle konkrétního typu zadání.
- Pokud v projektu nestihnete implementovat všechny požadované vlastnosti, je nutné veškerá omezení jasně uvést v dokumentaci a v souboru README.
- Co není v zadání jednoznačně uvedeno, můžete implementovat podle svého vlastního výběru. Zvolené řešení popište v dokumentaci.
- Při řešení projektu respektujte zvyklosti zavedené v OS unixového typu (jako je například formát textového souboru).
- Vytvořené programy by měly být použitelné a smysluplné, řádně komentované a formátované a členěné do funkcí a modulů. Program by měl obsahovat nápovědu informující uživatele o činnosti programu a jeho parametrech. Případné chyby budou intuitivně popisovány uživateli.
- Aplikace nesmí v žádném případě skončit s chybou SEGMENTATION FAULT ani jiným násilným systémovým ukončením (např. dělení nulou).
- Pokud přejímáte krátké pasáže zdrojových kódů z různých tutoriálů či příkladů z Internetu (ne mezi sebou), tak je nutné vyznačit tyto sekce a jejich autory dle licenčních podmínek, kterými se distribuce daných zdrojových kódů řídí. V případě nedodržení bude na projekt nahlíženo jako na plagiát.
- Konzultace k projektu podává vyučující, který zadání vypsál.
- Sledujte fórum k projektu, kde se může objevit dovysvětlení či upřesnění zadání.
- Před odevzdáním zkontrolujte, zda projekt obsahuje všechny potřebné soubory a také jste dodrželi jména odevzdávaných souborů pro konkrétní zadání. Zkontrolujte, zda je projekt přeložitelný.

## Hodnocení projektu:

- **Maximální počet bodů za projekt je 20 bodů.**
  - Maximálně 15 bodů za plně funkční aplikaci.
  - Maximálně 5 bodů za dokumentaci. Dokumentace se hodnotí pouze v případě funkčního kódu. Pokud kód není odevzdán nebo nefunguje podle zadání, dokumentace se nehodnotí.
- Příklad kritérií pro hodnocení projektů:
  - nepřehledný, nekomentovaný zdrojový text: až -7 bodů
  - nefunkční či chybějící Makefile: až -4 body
  - nekvalitní či chybějící dokumentace: až -5 bodů
  - nedodržení formátu vstupu/výstupu či konfigurace: -10 body
  - odevzdáný soubor nelze přeložit, spustit a odzkoušet: 0 bodů
  - odevzdáno po termínu: 0 bodů
  - nedodržení zadání: 0 bodů
  - nefunkční kód: 0 bodů
  - opsáno: 0 bodů (pro všechny, kdo mají stejný kód), návrh na zahájení disciplinárního řízení.

## Popis varianty:

### ZADÁNÍ:

Vášim úkolem je:

- 1) Nastudovat si protokol TFTP a sepsat relevantní dokumentaci.
- 2) Implementovat klientskou část tohoto protokolu, která by generovala žádosti na server a to jak pro IPv4 tak pro IPv6.
- 3) Otestovat a ukázat využití vašeho projektu v reálné situaci.

### UPŘESNĚNÍ ZADÁNÍ:

#### Ad 1)

K úspěšnému zvládnutí začněte číst standardy a drafty. Ideálně začněte čtením [IEN 133](#), které popisuje verzi 1, poté přejděte na verzi 2 (v její aktualizované podobě RFC [1350](#)), jež budete implementovat, a neopomeňte i na všechny další s problematikou související RFC [1785](#), [2090](#), [2347](#), [2348](#), [2349](#).

Dokumentaci lze realizovat v libovolném textovém procesoru a jazycích čeština, angličtina, slovenština. Při tvorbě dokumentace citujte v souladu s <https://www.fit.vut.cz/study/theses/citations/.cs>. Za přečtení stojí i <http://www.fit.vutbr.cz/~martinek/latex/citace.html>.

#### Ad 2)

Implementujte TFTPv2 klienta s rozšířeními timeout interval, blocksize option a transfer size option (ideálně i o multicast option). Klient bude vhodnou formou uvědomovat o své činnosti (nejlépe výpisem obsahujícím časové razítko, IP adresu klienta a rozbořem zprávy, která byla přijata/zpracována/zaslána), přičemž forma výstupu je volná. Klient po spuštění nabídne vlastní příkazovou řádku, která bude zpracovávat požadavky na přenos dle níže uvedených parametrů. Podpora mail-módu se nevyžaduje.

Projekt bude vypracován v jazyce C/C++. Měl by podporovat jak IPv4, tak IPv6 komunikaci. Využít lze standardní knihovny libpcap, libnet, BSD schranek, apod. Projekt by měl být přeložitelný a spustitelný na referenčním virtuálním stroji (pomocí aplikace [VMWare Player](#) nebo [VirtualBox](#)). Vyvíjet si můžete na libovolném systému, opravovat a hodnotit odevzdaný kód se však bude na referenčním. Vzhledem k nátuře projektu můžete předpokládat, že projekt bude spouštěn s rootovskými privilegii. Referenční image má přihlašovací údaje student/student s možnou eskalací privilegií jako root. Ke stažení je z těchto odkazů:

- [univerzitní OneDrive](#) (vanila VirtualBox edice)
- [univerzitní OneDrive](#) (VMWare 15+ edice)
- [NES fakultní server](#) (VMWare 15+ edice)

### **Závazný vstup a výstup:**

\$ myftpcient

>-R/W -d *adresar/soubor* -t *timeout* -s *velikost* -a *adresa,port* -c *mód* -m

- parametr -R nebo parametr -W (povinný)
  - Pokud bude zadán parametr -R, bude se jednat o čtení souboru ze serveru. Pokud bude zadán parametr -W, bude se jednat o zápis souboru na server. Přítomnost obou parametrů v rámci jednoho příkazu je vyloučena.
- parametr -d (povinný)
  - *soubor* určuje soubor, který bude klient z/na server přijímat/odesílat. *adresar* specifikuje absolutní cestu kam/odkud se má *soubor* přenést.
- parametr -t (nepovinný)
  - *timeout* v sekundách určuje, jaká je hodnota timeout, kterou bude klient poptávat od serveru. Klient navrhuje, server může či nemusí akceptovat. Samotný timeout vyjadřuje dobu čekání serveru před retransmisí nepotvrzených dat. Pokud není parametr přítomen, tak se option nepoužije.
- parametr -s (nepovinný)
  - *velikost* specifikuje maximální velikost bloku v násobcích oktětů, které bude klient navrhovat pro přenos. Jako horní hranici, kterou klient odesílá, uvažujte nejmenší z MTU ze všech rozhraní, na které by klient mohl TFTP zprávu serveru odeslat (pozor, tedy potenciálně více než 512 B, co je v RFC).
- parametr -m (nepovinný)
  - klient si od serveru vyžádá skrz multicast option transfer dat pomocí multicastu.
- parametr -c (nepovinný)
  - *mód*, který může nabýt jen hodnot "ascii" (nebo "netascii") a "binary" (nebo "octet"), čímž klient specifikuje mód přenosu dat. Implicitně (bez přítomnosti argumentu) se uvažuje binární transfer.
- parametr -a (nepovinný)
  - *adresa* je adresa serveru ve formátu IPv4 či IPv6 rozhraní, na němž má služba běžet (pokud není žádná uvedena, tak se implicitně uvažuje IPv4 localhost, tedy 127.0.0.1).
  - *port* je číslo portu, na kterém server naslouchá (pokud není uveden, je to implicitně 69). Pokud neuvedete parametr -a, uvažujte IPv4 127.0.0.1 a port 69. Pokud uvedete parametr -a, tak se držte

poptávané syntaxe *adresa,port*.

### Příklad spuštění:

```
$ myftpcient
> -R -d /home/user/readme.txt -a 147.229.181.2,169 -c octet
[2014-09-15 20:59:11.190] Requesting READ from server 147.229.181.2:69
[2014-09-15 20:59:11.454] Receiving DATA ... 1536 B of 2012545 B
...
[2014-09-15 20:59:18.454] Transfer completed without errors
>
...
$
```

### DOPORUČENÍ/OMEZENÍ:

- V dobré dokumentaci se OČEKÁVÁ následující: titulní strana, obsah, logické strukturování textu, výcuc relevantních informací z nastudované literatury (nikoli však Ctrl+C+V nad obsahem RFC či Wikipedie), popis zajímavějších pasáží implementace, bibliografie.
- Vlastní strategii při odmítnutí parametrů -t a -s serverem popište v Readme souboru a dokumentaci. Sofistikovanější řešení (jiné než ERROR pro celý transfer) budou lépe hodnocena.
- Zadávací prompt může být jak blokující (čeká se na dokončení předchozí operace, než se uživateli nabídne další možnost zadávání parametrů), tak konkurentní.
- Pořadí parametrů může být jakékoli. Mezi parametry, ale i oddělovači (čárka) může být libovolné nenulové množství bílých znaků. Pro zpracování parametrů lze použít *getopt()*.
- Všechny Vaše implementované programy by měly být POUŽITELNÉ a SMYSLUPLNÉ, řádně KOMENTOVANÉ pro použití dalšími programátory, případné výsledky a běh jejich činnosti pak INTUITIVNĚ interpretován uživateli!!!
- Do souboru Readme uvedete VE VLASTNÍM ZÁJMU případná omezení funkcionality vašeho projektu - na dokumentovanou chybu se nahlíží v lepším světle než na nedokumentovanou!
- Chybové hlášky patří na stderr, obvyklé výstupy běhu pak na stdout.
- Pro základní testování protistrany, můžete použít např. TFTP32 (<http://tftpd32.jounin.net/>).
- Mezi další referenční implementace patří třeba Linuxové originální <http://linux.die.net/man/8/tftpd> a <http://linux.die.net/man/1/tftp> či <https://linux.die.net/man/8/atftpd>
- Option multicast je bonusovým rozšířením projektu za až +4 body, kterými se budou moci lepit bové ztráty z jiných částí projektu. Na projekty s aspoň částečnou funkcionalitou v tomto směru bude nahlíženo lépe než na projekty, které ji nemají. Pro jednoduchost uvažujte, že klient je vždy a stále master klientem na segmentu.
- Transfersize option je přítomna v každém Requestu.
- MTU pro Blocksize option můžete získat např. z rozhraní pomocí *ioctl()*.

### LITERATURA:

- IEN 133
- RFC 1350
- RFC 1785

- RFC 2090
- RFC 2347
- RFC 2348
- RFC 2349
- <http://linux.die.net/man/8/in.tftpd>
- [http://en.wikipedia.org/wiki/Trivial\\_File\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol)