

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 však měl být přenositelný. Hodnocení projektů může probíhat na jiném počítači s nainstalovaným OS GNU/Linux, včetně jiných architektur než Intel/AMD, jiných distribucí, jiných verzí knihoven apod. 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 18.11.2019 (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.
- Před odevzdáním zkontrolujte, zda jste dodrželi všechna jména souborů požadovaná ve společné části zadání i v zadání pro konkrétní projekt. 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
 - odevzdaný 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:

Napište program `d6r`, který bude umět vložit do DHCPv6 zpráv MAC adresu klienta, jak definuje [RFC 6939](#), a logovat tyto informace pomocí protokolu `syslog`.

Spuštění aplikace

Použití: `d6r -s server [-l] [-d] [-i interface]`

Pořadí parametrů je libovolné. Popis parametrů:

- `-s`: DHCPv6 server, na který je zaslán upravený DHCPv6 paket.
- `-l`: Zapnutí logování pomocí `syslog` zpráv
- `-i`: Rozhraní, na kterém relay naslouchá, všechna síťová rozhraní, pokud parametr není definován.
- `-d`: Zapnutí debug výpisu na standardní výstup

Výstup aplikace

Na standardní výstup vypište informaci pouze pokud je zapnutý přepínač `-d`, ve formátu IPv6 adresa (prefix), MAC adresa klienta.

Ukázka možného výstupu:

```
$ d6r -s 2001:db8::1111 -d
2001:db8:a::/56,aa:bb:cc:dd:ee:ff
2001:db8:b::1,aa:bb:cc:dd:ff:ff
```

Upřesnění zadání

- Pro logování pomocí protokolu `syslog` využijte standardní volání knihovny `syslog`. Informaci logujte ve stejném formátu jako na standardní výstup, tj. prefix, MAC.
- Při vytváření programu je povoleno použít hlavičkové soubory pro práci se sokety a další obvyklé funkce používané v síťovém prostředí (jako je `netinet/*`, `sys/*`, `arpa/*` apod.), knihovny pro práci s vlákny (`pthread`), pakety (`pcap`), `syslog`, signály, časem, stejně jako standardní knihovnu jazyka C (varianty ISO/ANSI i POSIX), C++ a STL.

- Relay nemusí podporovat multicast komunikaci s DHCPv6 serverem.

Dokumentace:

Soubor Readme z obecného zadání nahraďte souborem d6r.1 ve formátu a syntaxi manuálové stránky - viz <https://liw.fi/manpages/>

Dokumentaci ve formátu pdf vytvořte dle pokynu v obecném zadání.

Doplňující informace k zadání

- Před odevzdáním projektu si důkladně pročtěte společné zadání pro všechny projekty.
- Jakékoliv rozšíření nezapomeňte zdůraznit v manuálové stránce a v dokumentaci. Není však možné získat více bodů, než je stanovené maximum.
- Program se musí vypořádat s chybnými vstupy.
- Veškeré chybové výpisy vypisujte srozumitelně na standardní chybový výstup.
- Pokud máte pocit, že v zadání není něco specifikováno, popište v dokumentaci vámi zvolené řešení a zdůvodněte, proč jste jej vybrali.
- Vytvořený kód by měl být modulární a otestovaný. Testy, které jste při řešení projektu napsali se spustí voláním "make test".
- Pište robustní aplikace, které budou vstřícné k drobným odchylkám od specifikace.
- Při řešení projektu uplatněte znalosti získané v dřívějších kurzech týkající se jak zdrojového kódu (formátování, komentáře), pojmenování souborů, tak vstřícnosti programu k uživateli.

Referenční prostředí pro překlad a testování

Program by měl být přenositelný. Referenční prostředí pro překlad bude server merlin.fit.vutbr.cz (program zde musí být přeložitelný). Vlastní testování může probíhat na jiném počítači s nainstalovaným OS GNU/Linux, včetně jiných architektur než Intel/AMD, jiných distribucí, jiných verzí knihoven apod. Pokud vyžadujete minimální verzi knihovny, jasně tuto skutečnost označte v dokumentaci a v manuálové stránce. Pro případné testování vaší aplikace lze využít virtuální stroj dostupný ke stažení [ZDE](#), informace k nainstalovaným balíčkůům a výchozím nastavení je na [githubu](#).

Doporučená literatura

- [RFC 6939](#)
- [RFC 8415](#)