**Společná část zadání projektu ISA**

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.

**Varianty zadání projektu**

* Varianty zadání projektu jsou uveřejněny v systému IS VUT. Přihlašování na varianty bude možné přes IS VUT od 26.9.2022 do 9.10.2022.
* Konzultace k zadání bude probíhat přes fórum v systému Moodle, viz níže.

**Pokyny k odevzdání projektu**

* Vypracovaný projekt uložený v archívu .tar a se jménem xlogin00.tar odevzdejte elektronicky přes IS VUT. Soubor nekomprimujte.
* **Termín odevzdání je 14.11.2022** **(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 ve formátu PDF (soubor *manual.pdf*), která bude obsahovat uvedení do problematiky, návrhu aplikace, popis implementace, základní informace o programu, návod na použití. Struktura dokumentace odpovídá technické zprávě a měla by obsahovat následující body: titulní stranu, obsah, logické strukturování textu včetně číslování kapitol, přehled nastudovaných informací z literatury, popis zajímavějších pasáží implementace, použití vytvořených programů a literatura. Pro dokumentaci lze použít upravenou šablonu pro [bakalářské práce](https://www.fit.vut.cz/study/theses/bachelor-theses/.cs).
  4. soubor *README*obsahující jméno a login autora, datum vytvoření, krátký textový popis programu s případnými rozšířeními či 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 vlastního uvážení. 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 vypsal. Pro své otázky můžete využít diskuzní fórum k projektům.
* 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 projektů**

* Hodnocení projektu bude zveřejněno  v IS VUT.
* **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 kriterií 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.

Generování NetFlow dat ze zachycené síťové komunikace – [Ing. Matěj Grégr, Ph.D.](https://www.vut.cz/lide/matej-gregr-84390)

V rámci projektu implementujte NetFlow exportér, který ze zachycených síťových dat ve formátu pcap vytvoří záznamy NetFlow, které odešle na kolektor.

**Použití:**  
Program musí podporovat následující syntax pro spuštění:  
  
./flow [-f <file>] [-c <netflow\_collector>[:<port>]] [-a <active\_timer>] [-i <inactive\_timer>] [-m <count>]

kde

-f <file> jméno analyzovaného souboru nebo STDIN,

-c <neflow\_collector:port> IP adresa, nebo hostname NetFlow kolektoru. volitelně i UDP port (127.0.0.1:2055, pokud není specifikováno),

-a <active\_timer> - interval v sekundách, po kterém se exportují aktivní záznamy na kolektor (60, pokud není specifikováno),

-i <seconds> - interval v sekundách, po jehož vypršení se exportují neaktivní záznamy na kolektor (10, pokud není specifikováno),

-m <count> - velikost flow-cache. Při dosažení max. velikosti dojde k exportu nejstaršího záznamu v cachi na kolektor (1024, pokud není specifikováno).

Všechny parametry jsou brány jako volitelné. Pokud některý z parametrů není uveden, použije se místo něj výchozí hodnota.

**Příklad použití:**

./flow -f input.pcap -c 192.168.0.1:2055

**Implementace:**

Implementujte v jazyku C/C++, za pomoci knihovny libpcap.

**Upřesnění zadání:**

* Jako export stačí použít NetFlow v5. Pokud byste implementovali v9 se šablonami, bude to bonusově zohledněno v hodnocení projektu.
* Pro vytváření flow stačí podpora protokolů TCP, UDP, ICMP.
* Informace, které neznáte (srcAS, dstAS, next-hop, aj.) nastavte jako nulové.
* Při exportování používejte původní časové značky zachycené komunikace.
* Pro testování můžete využít nástroje ze sady nfdump (nfdump, nfcapd, nfreplay, ...).
* Pro vytvoření vlastního testovacího souboru můžete použít program *tcpdump*.
* Exportované NetFlow data by měla být čitelná nástrojem nfdump.

**Odevzdání:**

Odevzdaný projekt musí obsahovat:

1. soubor se zdrojovým kódem,
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 *flow.1*ve formátu a syntaxi manuálové stránky - viz <https://liw.fi/manpages/>

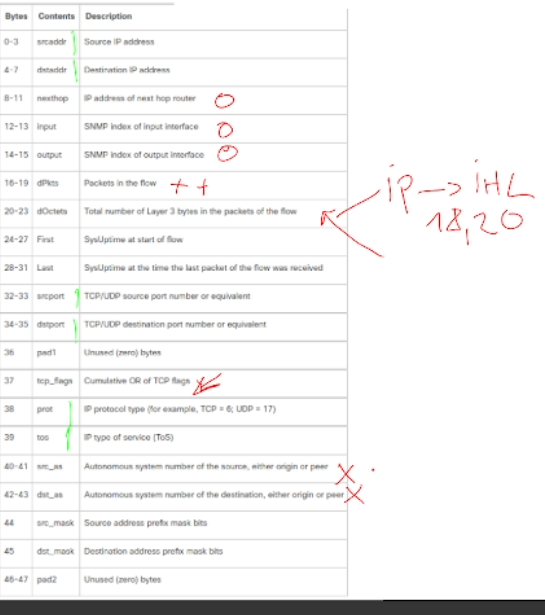
Vypracovaný projekt uložený v archívu .tar a se jménem xlogin00.tar odevzdejte elektronicky přes IS. Soubor nekomprimujte.

**Spuštění, testování:**

Všechny nezbytné úkony pro přípravu spuštění Vaší aplikace musí proběhnout zadáním příkazu **make,**ať už si vyberete kterýkoliv jazyk.

**Doporučená literatura:**

* Studijní materiály k předmětu ISA - NetFlow
* NetFlow na Wikipedia.org - <https://en.wikipedia.org/wiki/NetFlow>
* Formát NetFlow datagramu - <http://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html#wp1003394> [Table B-3 a Table B-4]
* man nfdump
* man nfcapd
* man libpcap

Text, letter

Description automatically generatedGraphical user interface, text

Description automatically generated