# Laboratorium 3
# Arytmetyka systemów komputerowych

## Informacje wstępne

- Celem laboratorium jest zidentyfikowanie podstawowych operacji wygenerowanych w assemblerze, w tym dodawania, odejmowania, mnożenia oraz dzielenia z użyciem różnych typów danych takich jak int, float oraz double.
- Dalsza część kursu zostanie zrealizowana z wykorzystaniem serwera obliczeniowego firmy Intel posiadającego dwa procesory Intel Xeon CPU E5-2699 v3 2.30GHz.
- Dostęp do serwerów jest realizowany w sposób zdalny i zostanie przydzielony na czas trwania kursu lub wybranych laboratoriów. Po zakończeniu kursu zostaną usunięte wszystkie pliki.
- Każdy uczestnik kursu nie może wykorzystywać zasobów obliczeniowych do innych celów aniżeli przewidziane do realizacji zadania w ramach przedmiotu Architektury Systemów Komputerowych.
- Wszyscy uczestnicy kursu będą wykonywać swoje zadania we wspólnej przestrzeni dyskowej używając jednego konta (użytkownika). Oznacza to, że każdy uczestnik kursu ma dostęp do plików innych uczestników, które może edytować i usuwać.
- Ze względu na powyższą formę pracy:
  - każdy uczestnik powinien realizować zadania we własnym folderze, zachowując regułę grzecznościową niezaglądania do folderów innych uczestników.
  - zachęcam do robienia regularnych kopii swoich plików/programów.

## Logowanie do systemu: Windows

- Przykładowo, aby uzyskać dostęp do serwera należy pobrać i zainstalować (a raczej uruchomić) terminal tekstowy *putty* oraz winscp:
  - https://www.putty.org/
  - https://winscp.net/eng/download.php
- Następnie, należy pobrać klucz prywatny umożliwiający autoryzację dostępu: http://icis.pcz.pl/~lszustak/ASK/laboratoria/key_windows.zip
  Po rozpakowaniu, w katalogu „key_windows” będzie plik o nazwie „*id_rsa.ppk*”, który należy zapisać w bezpiecznym miejscu. Uwaga! Klucz Został zabezpieczony hasłem: architektura
- Dodatkowe, informacje:
  - nazwa konta i nazwa hosta: phi20@miclab.pl
  - nr portu: 1
  - hasło do klucza: architektura
- Po zalogowaniu należy utworzyć własny katalog potrzebny do realizacji ćwiczeń:
  - mkdir -p wybrana_przez_ciebie_nazwa
- Przydatne komendy:
  - **ls** - wyświetla zawartość foldera
  - **pwd** - wyświetla ścieżkę, w której jesteśmy
  - **ctrl+l** - czyści ekran
  - **mkdir nazwa_folderu** - tworzy folder
  - **cd ścieżka_do_folderu** - umożliwia poruszanie się po przestrzeni dyskowej

## Logowanie do systemu: Linux

- W pierwszej kolejności należy pobrać klucz prywatny umożliwiający autoryzację dostępu:
  http://icis.pcz.pl/~lszustak/ASK/laboratoria/key_linux.tar
- Należy się upewnić, że folder .ssh istnieje. Jeżeli folder .ssh nie istnieje, należy go utworzyć komendą: **mkidr ~/.ssh/**
- Następnym krokiem jest przeniesienie pobranego klucza do folderu .ssh. W tym celu wykorzystujemy komendę: **mv ~/Pobrane/key_linux.tar ~/.ssh/**
  **Uwaga:** Należy się upewnić, że podano poprawną ścieżkę do klucza.
- Przeniesiony plik z kluczem należy rozpakować. W tym celu przechodzimy do folderu, w którym się znajduje komendą: **cd ~/.ssh/** Następnie wykonujemy polecenie: **tar -xf key_linux.tar**
- Po rozpakowaniu, w bieżącym katalogu znajdują się dwa pliki niezbędne do logowania: id_rsa oraz id_rsa.pub

- Aby zalogować się na serwer należy skorzystać z protokołu ssh z poziomu terminala:
  **ssh -i id_rsa phi20@miclab.pl -p 1**
- Uwaga! Klucz został zabezpieczony hasłem: **architektura**
- Po zalogowaniu należy utworzyć własny katalog potrzeby do realizacji ćwiczeń:
  - **mkdir -p wybrana_przez_ciebie_nazwa**
- Przydatne komendy:
  - **ls** - wyświetla zawartość foldera
  - **pwd** - wyświetla ścieżkę, w której jesteśmy
  - **ctrl+l** - czyści ekran
  - **mkdir nazwa_folderu** - tworzy folder
  - **cd ścieżka_do_folderu** - umożliwia poruszanie się po przestrzeni dyskowej

- Transfer plików pomiędzy komputerem a serwerem:
  - W pierwszej kolejności należy upewnić się z jakiej dystrybucji Linuxa korzystamy. W tym celu z terminala wykonujemy polecenie: **lsb_release -a** W salach laboratoryjnych dostępny jest Ubuntu oraz CentOS.
  - Następnie otwieramy dowolny folder i w miejsce ścieżki wpisujemy:
    - **ssh://phi20@miclab.pl/home/phi20/** w przypadku CentOS
    - **fish://phi20@miclab.pl/home/phi20/** w przypadku Ubuntu
  - Po wprowadzeniu hasła do klucza otrzymujemy dostęp do plików znajdujących się na serwerze. Uwaga: Należy korzystać tylko z własnego folderu.

## Zadanie 1

- Utwórz plik w swoim folderze o unikalnej nazwie np. ***lab3.cpp***, a następnie wklejć poniższy program:

```
#include <iostream>

int main()
{

  float a = 5;
  float b = 4;
  float c = 0;

  c = a + b;

  std::cout<<c<<std::endl;

  return 0;
}
```

- Skompiluj utworzony program tak aby wygenerować kod do asemblerze korzystając z następującej komendy: **g++ -S *lab3.cpp* –O0**
- Otwórz wygenerowany plik o nazwie ***lab3*.s** w celu znalezienia fragmentu kodu odpowiedzialnego za wykonanie operacji dodawania (szukaj linijki kodu zawierającej słowa „*add*”), a następnie „zapamiętaj/zapisz” instrukcję.

```
movl    $0, -4(%rbp)
movl    -12(%rbp), %edx
movl    -8(%rbp), %eax
addl    %edx, %eax
movl    %eax, -4(%rbp)
movl    -4(%rbp), %eax
```

- W kolejnych krokach:
  - zamień w głównym pliku ***lab3*.cpp** operację dodawania na inne operacje np. odejmowania, mnożenia i dzielenia, skompiluj, znajdź odpowiednią instrukcję i ją zapamiętaj
  - powtórz powyższe czynności dla różnych typów danych w tym int, float oraz double, jak również użyj operacji mnożenia i dzielenia dla operacji zmiennoprzecinkowych.

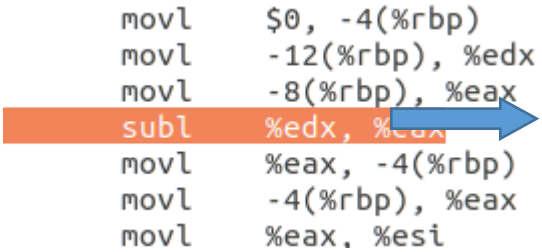
| | **int** | **float** | **double** |
|---|---|---|---|
| **operacja +** | addl | **.....** | **.....** |
| **operacja -** | **.....** | **.....** | **.....** |
| **operacja *** | **.....** | **.....** | **.....** |

| operacja / | ..... | ..... | ..... |
|---|---|---|---|

**Zadanie 2**

Korzystając z powyższej tabelki, dla wybranej kombinacji z zadania 1 (np. operacja dodawania dla typu int), edytuj plik wygenerowany w assemblerze, znajdź linijkę kodu odpowiedzialną za operację np. dodawania i zamień komendę dodawania na komendę asemblera odejmowania.

```
movl    $0, -4(%rbp)
movl    -12(%rbp), %edx
movl    -8(%rbp), %eax
addl    %edx, %eax
movl    %eax, -4(%rbp)
movl    -4(%rbp), %eax
```




Zapisz plik, skompiluj i uruchom w następujący sposób:

- kompilacja: **g++ *lab3*.s –O0 –o exe**
- uruchomienie: **./exe**

Sprawdź czy zamiana instrukcji dodawania na instrukcję odejmowania zadziałała zgodnie z oczekiwaniem.

Powtórzy powyższą czynność dla różnych typów danych podmieniając w asemblerze różne operacje:

- Kombinacja: operacja dodawania na typie int oraz zamiana na operacje -,*,/
- Kombinacja: operacja dodawania na typie float oraz zamiana na operacje -,*,/
- Kombinacja: operacja dodawania na typie double oraz zamiana na operacje -,*,/