25 Programování v JSA

**Binární kód**: Binární kód je způsob reprezentace dat a instrukcí v počítači pomocí binárních čísel (0 a 1). Tyto binární čísla jsou interpretována hardwarem počítače jako instrukce nebo data.

**Jazyk symbolických adres**: Jazyk symbolických adres je způsob zápisu programů, který umožňuje použití symbolických názvů pro paměťové adresy a instrukce místo přímého použití binárního kódu. To zjednodušuje psaní a porozumění programům.

**Assembler**: Assembler je program, který překládá zdrojový kód napsaný v jazyce symbolických adres do ekvivalentního strojového kódu (binárního kódu), který může počítač přímo vykonávat.

**Překlad programu**: Překlad programu je proces převodu zdrojového kódu napsaného v programovacím jazyce do strojového kódu, který je počítačem přímo vykonatelný. Tento proces může zahrnovat různé fáze, včetně analýzy, syntaktického a sémantického zpracování a generování strojového kódu.

**Zápis programu v JSA**:

Pole návěští – slouží pro orientaci v programu, jsou na něj odkazovány instrukce skoku. Identifikátor návěští se ukončuje dvojtečkou

Pole komentáře – komentář je v JSA důležitý, bez něj se v programu skoro nedá orientovat. Komentář se píše za středníkem

**Instrukce**: Instrukce v JSA mohou zahrnovat instrukce přesunu (např. MOV), aritmetické instrukce (např. ADD, SUB), logické instrukce (např. AND, OR), instrukce posuvu a rotací, instrukce skoku (např. JMP, JZ), řídící instrukce (např. CALL, RET) a další specifické instrukce podle konkrétní architektury procesoru a programovacího jazyka.

Příklad sčítání

.equ cislo1 = 10

.equ cislo2 = 20

Mov ax,cislo1

Mov bx,cislo2

Add ax,bx