

**Storcz Tamás**

Senior Fejlesztő  
Crosssec Solutions Kft.

---

# Alapismertetek

Alacsonyszintű programozás

# CPU FELÉPÍTÉSE I

## Control Unit

Ez szervezi, ütemezi a processzor egész munkáját. Például lehívja a memóriából a soron következő utasítást, értelmezi és végrehajtatja azt, majd meghatározza a következő utasítás címét.

## Register

A processzorba beépített nagyon gyors elérésű, kis méretű memória elemek. Addig tárolják az információt, amíg a processzor dolgozik velük.

- *Utasításszámláló*: (Program Counter, Instruction Pointer) tárolja a következő végrehajtandó utasítás címét
- *Utasításregiszter* (Instruction Register) a végrehajtandó utasítást tárolja
- *Flagregiszter*: a processzor működése közben létrejött logikai állapotjelzők értékeit tárolja,
- *Veremmutató (Stack Pointer)*: A veremmutató regiszter a verem legfelső elemét jelöli ki.
- *Akkumulátor* (Accumulator) a logikai és aritmetikai műveletek egyik operandusát, és az eredményt tartalmazza.
- *Általános célúregiszterek*: adattárolók, adat manipulációs műveletek másik operandusa

# CPU FELÉPÍTÉSE II

## Arithmetic and Logical Unit

Számítási (aritmetikai és logikai), ill. összehasonlító műveleteket hajt végre a tárolt adatokon

## Floating Point Unit

*Lebegőpontos műveleteket végző egység*

## Address Generation Unit

Feladata a programutasításokban található címek leképezése a főtár fizikai címekre és a tárolóvédelmi hibák felismerése.

## Buszvezérlő

A regisztert és más adattárolókat összekötő buszrendszert irányítja.

## Cache

A cache a processzorba, vagy a processzor környezetébe integrált memória, ami a viszonylag lassú rendszermemória-elérést hivatott kiváltani azoknak a programrészeknek és adatoknak előzetes beolvasásával, amikre a végrehajtásnak közvetlenül szüksége lehet.

# A PROGRAM

## Mi a program?

**Fizikailag:** bithalmaz - sokféle fizikai reprezentációs lehetőség

### Logikailag

Készítés – forráskód

*Programozónak: Változók, metódusok, objektumok, tulajdonságok, interfészek*

Felhasználás – futtatható

*Számítógépnek: A vérehajtandó utasítások, adatok (operációk, operandusok)*

*Felhasználónak: Megoldás egy konkrét feladatra*

---

# EGY PROGRAM ELEMEI

## Adatok

A program írója által előre, vagy felhasználója által működés közben meghatározott numerikus adatok.

## Manipulációs utasítások

Az adatok manipulálására szolgáló utasítások (mozgatás, aritmetikai és logikai műveletek)

## Vezérlő utasítások

A végrehajtandó utasítások sorrendjét befolyásolják

---

# VEZÉRLÉSI SZERKEZETEK

## SZEKVENCIA

Legegyszerűbb vezérlési szerkezet. Az utasítások egymás után történő végrehajtása.

## SZELEKCIÓ

Egy feltétel kiértékeléséből eredő logikai érték alapján a következő utasítás kihagyása.

## UGRÁS

Az utasítások végrehajtásának folytatása (előre meghatározott) tetszőleges helyen

## ITERÁCIÓ

Egy utasításblokk ismételt végrehajtása. (számláló, elöl/hátul tesztelő)

## SZUBRUTIN HÍVÁS

Egy speciális, összetett feladat elvégzésére készített utasítás gyűjtemény (függvény/eljárás)

---

# VÁLTOZÓK

## TÍPUS

A változó által lefoglalt memória terület tartalmának értelmezéséhez szükséges információ

## ÖSSZETETT TÍPUS

Több, egyszerű típusú adatból álló tartalom azonosítója

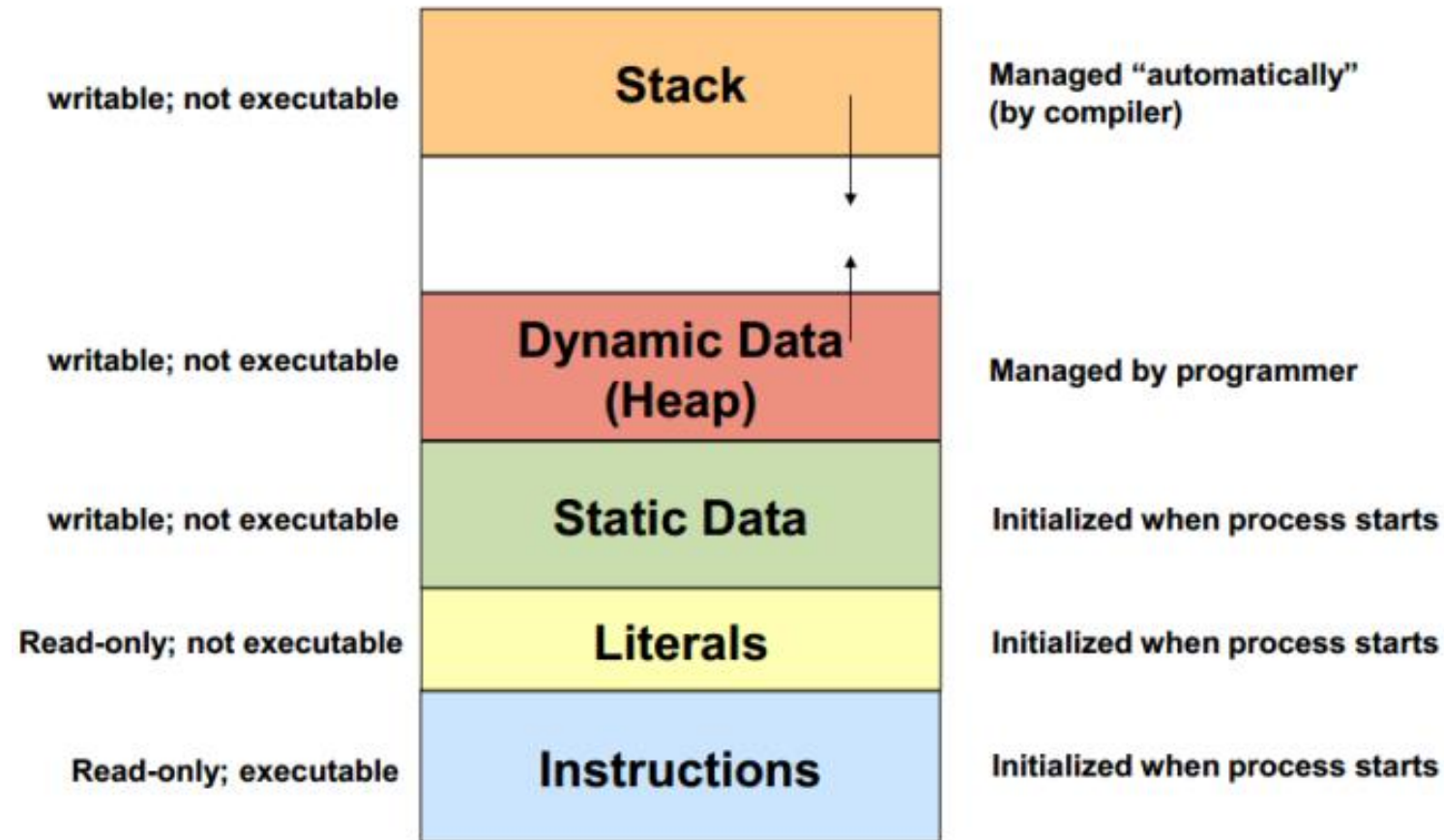
## POINTER TÍPUS

Egyszerű típus, memóriacímet tartalmaz

## ÉRTÉK ÉS REFERENCIA

Az érték típusú változó közvetlenül az értéket tárolja, a referencia típusú pedig az értékre történő hivatkozást. Jelentősége a paraméter átadásnál és a dinamikus memóriakezelésnél van.

# MEMÓRIAKEZELÉS





# COMPILE - FORDÍTÁS

A **computer** olyan program, amely képes egy forráskódot másik nyelvre (pl. gépi kód) lefordítani.

## FORDÍTÁSI MŰVELET

A forráskód fájlok feldolgozását és „object” fájlok előállítását jelenti. Ebben a lépésben nem állítunk elő semmilyen futtatható állományt, ehelyett a forráskódnak megfelelő gépi utasítások sorozatának valamilyen reprezentációját készítjük el.

## LÉPÉSEI

Lexikális elemzés, Színtaktikai elemzés, Szemantikai elemzés, Optimalizálás, Kód generálás

---

# LINK - SZERKESZTÉS

Feladata a fordítási műveletekkel előállított **object** fájlok futtatható állományba szervezése.

A **linker** elkészíti a futtatható állomány belépési pontját, ellenőrzi a hivatkozott eljárások, függvények meglétét. A **linker** nem csak egy fájlban dolgozik, feladata a különböző **object** fájlok közötti hivatkozások feloldása.

---

# A PROGRAM

A fordítási és linkelési lépéseink kimenete tehát egy olyan program, amely adat manipulációs és vezérlő utasítások véges sorozata. Ezt a számítógép Determinisztikus Véges Automataként végrehajtja. (állapotai és azok átmenetei előre meghatározottak, van kezdeti és végállapot)

Ez a végrehajtás nem intelligens. Csak az aktuális (és esetleg a néhány következő) utasítás számít.

**A végrehajtásra a forráskódunk struktúrája NINCS hatással.**