


Проф. д-р П. Митревски

Архитектура и организација на компјутери



Фонд на часови: **3+2** (летен семестар)

Препорачана литература:

- A. S. Tanenbaum: "Structured Computer Organization", 5th edition, Prentice Hall, ISBN: 0-13-148521-0
- J. Hennessy, D. Patterson: "Computer Architecture: A Quantitative Approach", 5th edition, Morgan Kaufmann Publishers, ISBN: 0-12-383872-X

Проф. д-р П. Митревски

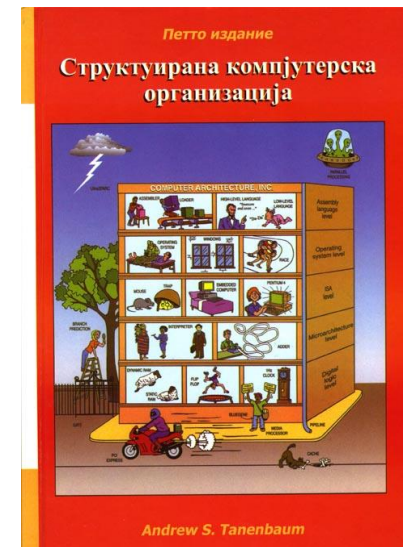
Архитектура и организација на компјутери

Ендру С. Таненбаум

„Структурирана компјутерска организација“
(петто издание)

ISBN: 978-608-4597-03-2

(превод на книга од која се учи на врвните универзитети во Европа и во светот)





1 Вовед

- Компјутерот е **машина** која решава проблеми извршувајќи инструкции што и' се дадени од човекот
- Низа од инструкции кои опишуваат **како** треба да се изврши определена задача, се нарекува **програма**
- Електронските кола на кој било компјутер се во состојба да препознаваат и директно да извршуваат само ограничено множество едноставни инструкции:
 - Собери два броја
 - Провери дали некој број е нула
 - Копирај податоци од една на друга локација во компјутерската меморија



1 Вовед

- Сите примитивни инструкции, заедно, сочинуваат јазик со чија помош човекот може да комуницира со компјутерот, наречен **машински јазик**
- Обично, се настојува примитивните инструкции да бидат колку што е можно поедноставни, со цел да се намали комплексноста и цената на чинење на неопходната електроника

1.1 Јазици, нивоа и виртуелни машини

- Постои голем расчекор помеѓу тоа што е погодно за човекот и тоа што е погодно за компјутерот – човекот сака да прави X, но компјутерот може да прави само Y
(👉 ПРОБЛЕМ!)
- Проблемот е решлив на два начина – но и во двата случаи треба да се дизајнира ново множество инструкции кое е многу погодно за употреба од страна на човекот, отколку множеството вградени машински инструкции
- Новите инструкции, заедно, сочинуваат јазик L1, исто како што вградените машински инструкции сочинуваат јазик L0

1.1 Јазици, нивоа и виртуелни машини

■ Преведување (translation) –

- Најпрво, секоја инструкција од програмата напишана во јазикот L1 се заменува со соодветна низа од инструкции од јазикот L0
- Потоа, компјутерот ја извршува новата L0-програма, наместо старата L1-програма

■ Интерпретација (interpretation)

- Постои програма која е претходно напишана во јазикот L0, наречена **интерпретатор (interpreter)**
- Интерпретаторот ги прегледува (една по една) инструкциите од програмата напишана во јазикот L1 и непосредно (веднаш) ги извршува соодветните низи од L0-инструкции

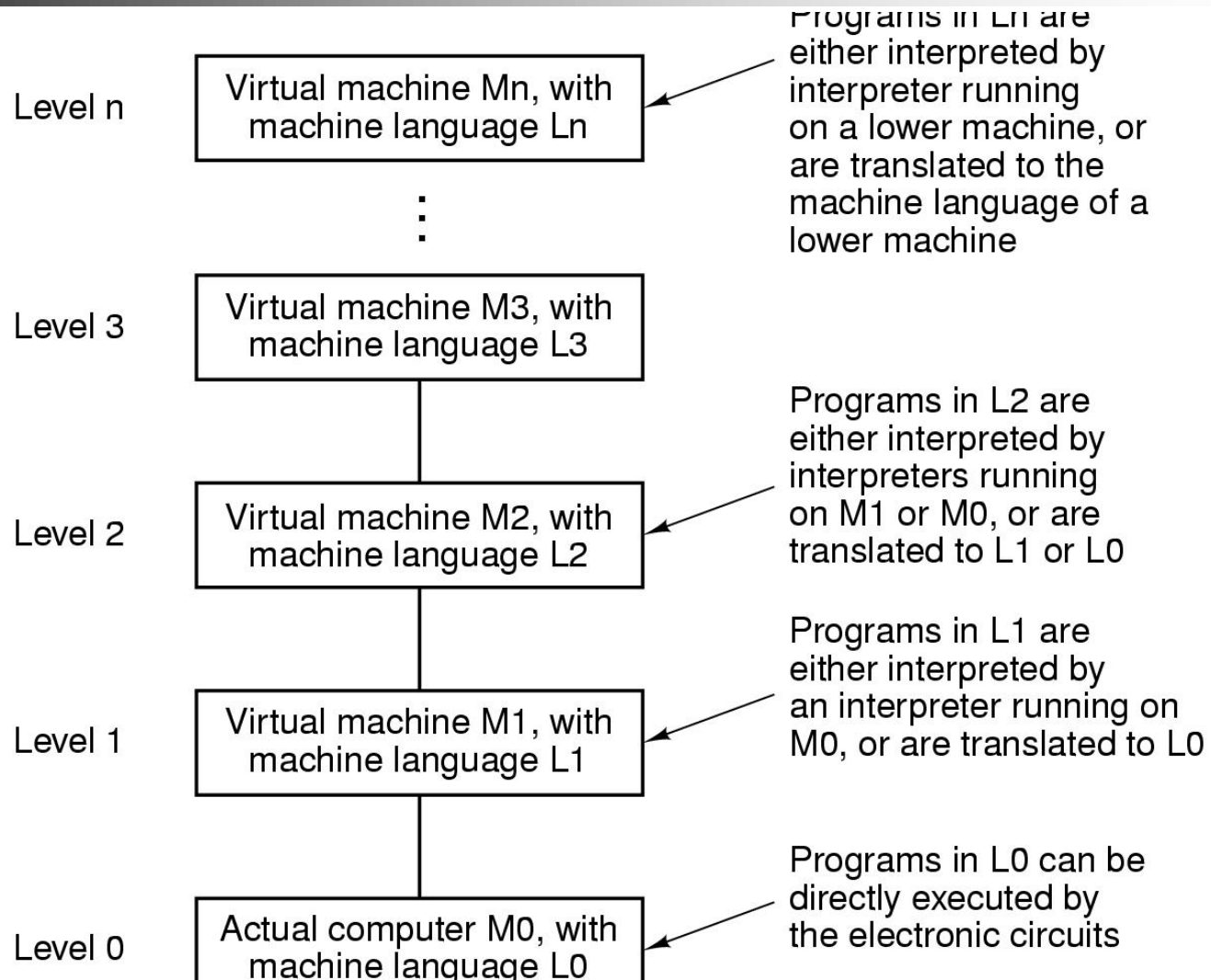
1.1 Јазици, нивоа и виртуелни машини

- Наместо да се размислува за преведување или интерпретирање, често пати е поедноставно да се замисли дека постои хипотетички компјутер – **виртуелна машина** (M1), чиј машински јазик е јазикот L1. Тогаш, човекот може да пишува програми за виртуелната машина, исто како таа да постои во стварноста
- За да биде преведувањето или интерпретирањето практично применливо, јазиците L0 и L1 не смеат „многу“ да се разликуваат – иако подобар од L0, јазикот L1 сепак ќе биде далеку од идеален за повеќето апликации

1.1 Јазици, нивоа и виртуелни машини

- Може да се дефинира цела серија јазици, секој од нив попогоден од неговите претходници... и така до недоглед(!)
- Бидејќи секој од јазиците се потпира на неговиот претходник, компјутерот може да се набљудува како серија од **Нивоа (layers, levels)**, поставени едно врз друго, од наједноставното до најсофистицираното
- Секоја машина има свој машински јазик кој го сочинуваат сите инструкции што таа може да ги извршува – **машината го определува јазикот** и, обратно, **јазикот ја определува машината**

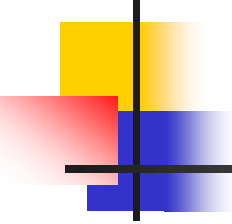
1.1 Јазици, нивоа и виртуелни машини



1.1 Јазици, нивоа и виртуелни машини



- Лицето чија задача е да пишува програми за виртуелна машина од ниво **n**, не мора воопшто да се интересира за преведувачите или интерпретаторите кои стојат подолу – структурата на машината гарантира дека програмите, на каков-таков начин, ќе можат да се извршуваат
- Повеќето програмери се интересираат само за најгорното ниво – она кое најмалку потсеќа на машинскиот јазик на дното



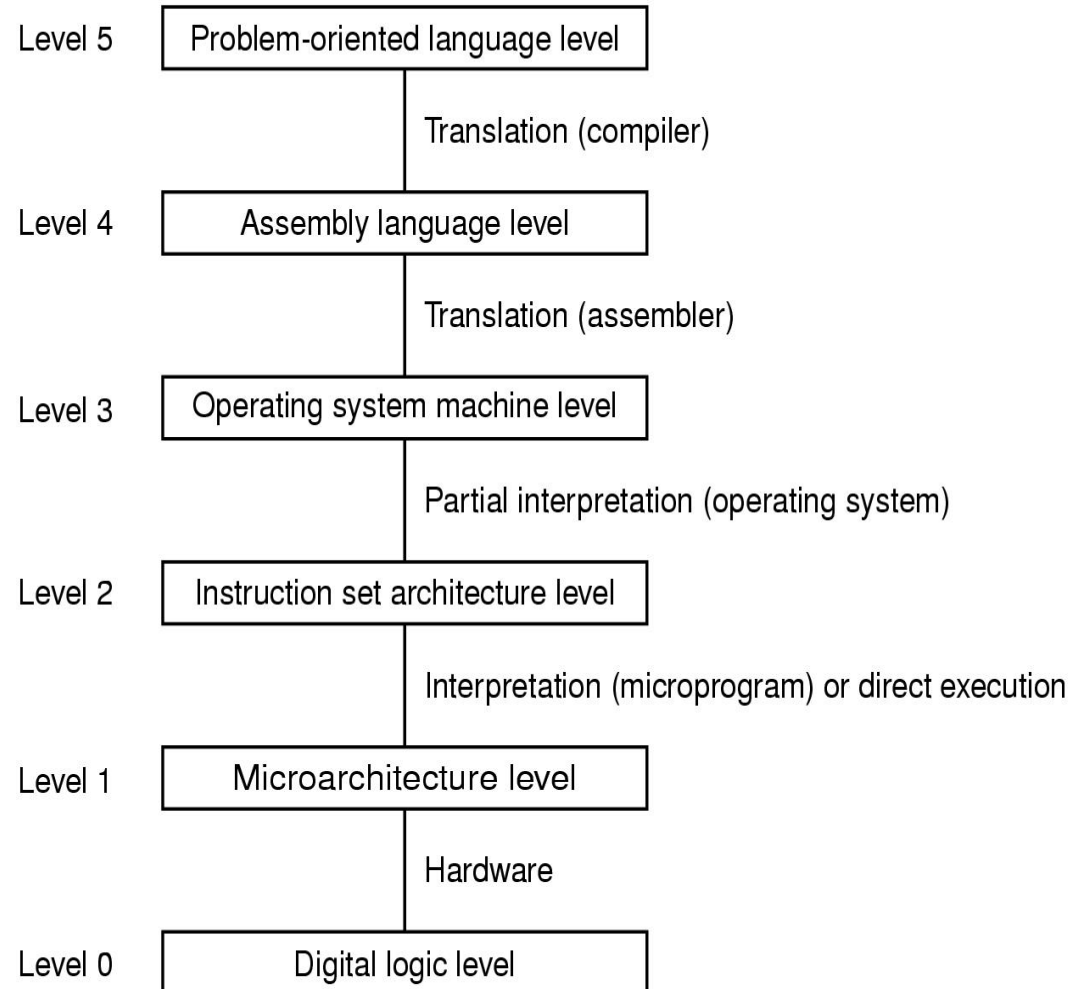
1.1 Јазици, нивоа и виртуелни машини

- Лицата чија цел е да разберат **како** компјутерот навистина работи, мораат да ги изучуваат сите нивоа
- Во продолжение, ќе стане збор за:
 - **коцептите и техниките за конструирање на пресметувачки машини (компјутери) како серија од нивоа**
 - **деталите на секое од нив(!)**

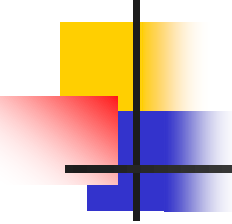
1.2 Современи машини со повеќе нивоа

- Повеќето современи компјутери се состојат од две или повеќе (па дури и шест!) нивоа:

- Проблемски-ориентирани јазици
- Асемблерски јазик
- Оперативен систем
- Архитектура на инструкциското множество
- Микроархитектура
- Дигитална логика

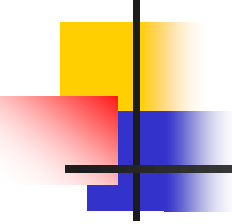


1.2.1 Ниво на дигиталната логика (L0)



- Елементи од интерес се **порти (gates)**
- Иако се составени од аналогни компоненти (транзистори), можат да се моделираат како дигитални уреди
- Секоја порта има еден или повеќе дигитални влезови (сигнали кои претставуваат 0 или 1), а како излез пресметува некоја едноставна функција од влезовите, како (на пр.) **AND** или **OR**

1.2.1 Ниво на дигиталната логика (L0)



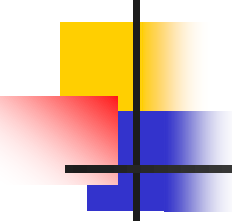
- Одреден број порти можат да се комбинираат и да формираат 1-битна меморија, која може да чува една единствена 0 или 1
- 1-битните мемории можат да се комбинираат во групи од (на пр.) 16, 32, или 64 и да формираат **регистри (registers)**
- За секој регистер велиме дека чува **еден единствен** бинарен број
- Портите, исто така, можат да се комбинираат за да ја формираат самата пресметувачка машина (engine)

1.2.2 Ниво на микроархитектурата (L1)

- Група од (типично) 8 до 32 регистри сочинуваат некаква **локална меморија (register file)**
- Посебно коло се нарекува **аритметичко-логичка единица (Arithmetic Logic Unit – ALU)**, која е способна да изведува едноставни аритметички и логички операции
- Регистрите се поврзани со ALU и формираат **податочна патека (data path)**
- Основната операција на податочната патека се состои од избирање на еден или два регистри над кои ќе биде извршена некоја аритметичко-логичка операција, при што резултатот повторно ќе биде запишан во некој регистер

1.2.2 Ниво на микроархитектурата (L1)

- Кај некои машини, функционирањето на податочната патека е контролирано од посебна програма наречена **микропрограма (microprogram)**, а кај други директно од хардверот
- Кај машините со софтверска контрола на податочната патека, микропрограмата е, всушност, интерпретатор за инструкциите од погорното ниво L2



1.2.3 Ниво на архитектурата на инструкциското множество (L2)

- Вообичаено, производителите на компјутери издаваат некаков прирачник со наслов “Machine Language Reference Manual” (или сл.), во кој се опишува **инструкциското множество**, односно множеството инструкции кои можат да бидат извршени ИЛИ директно од хардверот ИЛИ по пат на интерпретација од страна на микропрограмата

1.2.4 Ниво на оперативниот систем (L3)

- Повеќето инструкции од јазикот на ова ниво се, исто така, присутни и во нивото L2
👉 (не постои причина поради која некоја инструкција која се појавува во едно ниво не би смеела да биде присутна и во други нивоа)
- Но, постои дополнително множество нови инструкции, различна мемориска организација, способност за конкурентно извршување на две или повеќе програми, ИТН.

1.2.4 Ниво на оперативниот систем (L3)

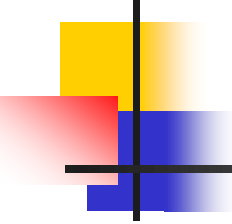
- Новите функционалности, карактеристични за нивото L3, ги овозможува интерпретатор кој се извршува на ниво L2, наречен **оперативен систем**
- Некои од L3 инструкциите ги интерпретира оперативниот систем
- Но, инструкциите кои се идентични со оние од нивото L2, ги интерпретира микропрограмата, или се извршуваат директно од хардверот
- Затоа, L3 се нарекува **хибридно ниво**

1.2.5 Ниво на асемблерски јазик (L4)

- За разлика од нивоата L2 и L3 кои **секогаш** се интерпретираат, нивоата L4 и L5, **обично** (но не секогаш), се преведуваат
- За разлика од јазиците на нивоата L1, L2 и L3 кои се **нумерички** (нули и единици), почнувајќи од нивото L4, јазиците содржат зборови и кратенки со какво-такво значење за човекот

1.2.5 Ниво на асемблерски јазик (L4)

- Асемблерскиот јазик (ниво L4) е **симболичка форма** за некој од јазиците од пониските нивоа – човекот е во можност да пишува програми за нивоата L1, L2 и L3 во поприфатлива форма, користејќи т.н. **мнемонички кодови** (ADD, SUB, MUL, DIV...)
- Програмите напишани во асемблерски јазик, најпрво се преведуваат во јазик од ниво L1, L2 или L3, а потоа се интерпретираат од соодветната виртуелна или вистинска машина
- Програмата што го врши преведувањето се вика **асемблер (assembler)**



1.2.6 Ниво на проблемски-ориентирани јазици (L5)

- Нивото L5 го сочинуваат јазици наменети за апликативните програмери – **јазици од високо ниво (High-Level Languages, HLL)**: BASIC, C, C++, Java, LISP, Prolog, ...
- Програмите напишани во овие јазици, најчесто се преведуваат до ниво L3 или L4 со помош на **преведувачи (compilers)**, иако, понекогаш, се интерпретираат (на пр. програми напишани во Java)



Резиме

- Множеството податочни типови, операции, и карактеристики на секое од нивоата, се нарекува негова **архитектура (на тоа ниво)**
- Архитектурата се занимава само со оние аспекти кои се **видливи** за корисникот на соодветното ниво
 - 👉 Она што го интересира програмерот, е **колку** меморија има на располагање...
- Другите аспекти на имплементацијата, како, на пример, **каква** технологија е искористена за изработка на меморијата, **не** се дел од архитектурата
- Изучувањето на тоа **како** да се дизајнираат оние делови од компјутерскиот систем кои се видливи за програмерите, се нарекува **архитектура на компјутери** (организација на компјутери)