

BLM 4811 MESLEKİ TERMİNOLOJİ – II

Ders 3:

Yazılım ve Programlama Dillerinin Gelişimi

Herşey Nasıl Başladı?

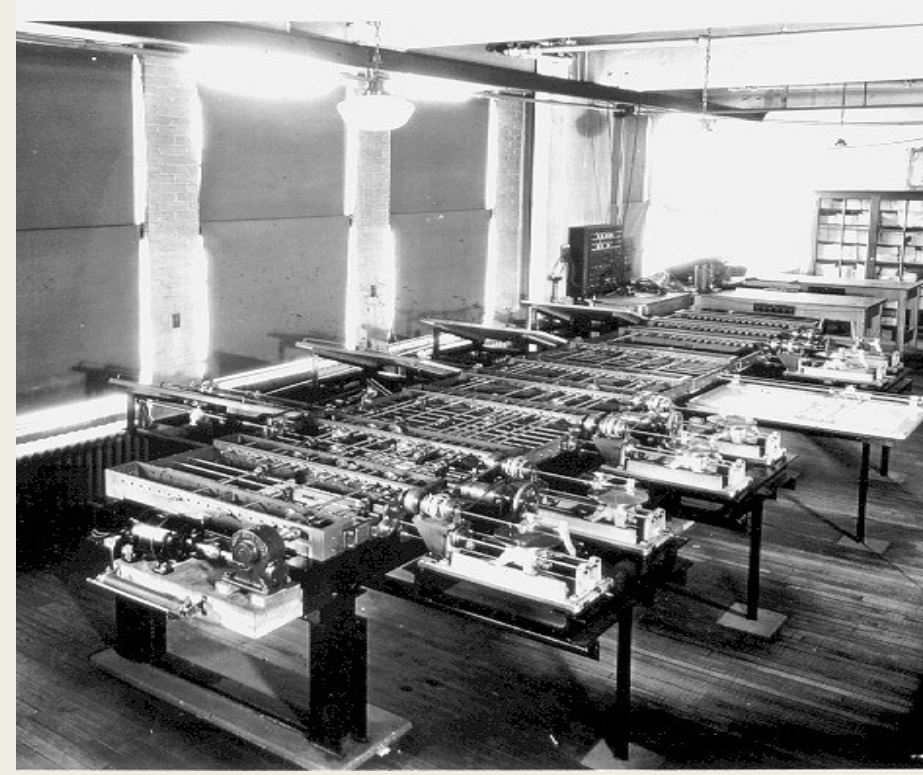
Dokuma Tezgahı – Joseph Marie Jacquard (1804)

- İlk tam otomatik ve PROGRAMLANABİLİR dokuma tezgahı
- Kumaş olarak dokunacak deseni delikli kartlar ile programlayabiliyordu.
- Jacquard'ın keşfi, elektro-mekanik ve elektronik hesaplamada veri girdi/çıktısı için uygun bir model oluşturdu, IBM temel olarak kullandı.



Analitik Makine

- Charles Babbage 1837’de başladı, 1871’de ölene kadar üzerinde çalıştı
- Delikli kartlarla sıralı kontrol, döngü, koşullu dallanma özelliklerini sağlıyordu.
- Kartlarda hem veri, hem de komut taşıyabiliyordu.
 - *İki ana parça: Store (Hafıza) ve Mill (işlemci) Hafızada numaralar saklanıyor, işlemcide yeni sonuçlara “dokunuyordu”.*
 - *Modern bilgisayarlarda: Hafıza ve işlem ünitesi (Memory – CPU)*



Augusta Ada Byron King, Lovelace Kontesi (1815-1852)

- Romantik şair Lord Byron'un kızı
- “Üretilmemiş” Analitik makine için Bernoulli sayılarını hesaplayan ilk “bilgisayar programını” 1843 yılında yazdı.
- Aslında ilk algoritmayı tasarlamıştı.
- İlk alt programı ve ilk döngüyü yarattı.
- Babbage'ın aksine, tüm çalışmalarını belgeleriyle kayıt altına aldı.



Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

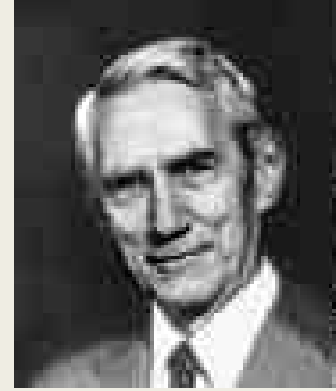
[illegible]

John von Neumann

- Bilgisayarın hafıza biriminin bir kısmını programı depolamak için kullanmayı tasarladı.
 - *Paylaşılan program tekniği*
- Böylece bilgisayar komutları kendi hafızasından alabilecekti. Dış kablolama ihtiyacını ortadan kaldırdı: EDVAC
- Şartlı kontrol transferi: alt programlara olanak verdi.
 - *Bilgisayar mantıksal karşılaştırma sonucuna göre farklı kod çalıştırabilir, bir kodu tekrar tekrar çalıştırabilir.*
 - *Kod blokları tekrar tekrar kullanılabilir.*



Claude Shannon (1916-2001)



- Information Theory'nin babası
 - *Matematiksel kuramlarla Mühendislik İlkelerini bir araya getirerek sayısal bilgisayarların üretilebilmesini sağladı.*
 - *A Mathematical Theory of Communication*
- Short Code dili: 1949
 - *0-1 değişimleri elle yapılıyor.*

Grace Hopper

- Mark I içinde bir ölü güve bularak, tarihteki ilk bug'ı buldu!
- 1951'de ilk derleyici olan A-0'ı geliştirdi.
 - *Dilin ifadelerini bilgisayar diline çeviriyordu.*
- Sonra COBOL'u oluşturan ekipte yer aldı.



9/9

0800

Antan started

1000

stopped - antan ✓

13" MC (032) MP - MC

(033) PRO 2

convd

Relays 6-2 in 033 failed special speed test
in relay

Relays changed

1100

Started Cosine Tap (Sine check)

1525

Started Mult + Adder Test.

1545

Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

1630

Antan started.

1700

closed down.

{ 1.2700 9.037 847 025

9.037 846 995 convd

~~1.982647000~~~~2.130476415~~ 4.615925059(-2)

2.130476415

2.130676415

10.000 test -

İşletim Sistemleri 1960'lar

- Merkezi bilgisayarların gelişmesi ile birlikte programların bilgisayarda çalıştırılmalarını yönetmek gerekiyordu
 - *İşletim sistemi, çevre birimleri ve diğer kaynakların yönetilmesini ve kullanılmasını sağlayan uygulamalar bütünüdür*
 - *Zaman paylaşımı ve düzenlemesi özelliği ile kullanıcılar, farklı işlemlerde bilgisayarı paylaşabiliyordu*
 - *Bilgisayarlar küçük boyutlu işletmelerce ulaşılabilir oldukça, özelleşmiş programlama dilleri ortaya çıktı*
 - Pascal (1971, Wirth), C (1972, Ritchie)

İlk Bilgisayar Oyunu, İlk kelime işlemci - 1962

- DEC PDP-1
üzerinde
MIT'den Steve
Russell
“Spacewar”
- TECO (Text
Editor and
Corrector)
- İlk kelime
işlemci
programı, Steve
Piner ve Peter
Deutsch, MIT



Kişisel hesaplama – 1970 sonları

- *Gates & Allen Microsoft'u 1975'te kurdu*
 - Gates kişisel bilgisayar için BASIC derleyicisini yazdı
 - Dünya yazılım devine dönüştü, Gates dünyanın en zengin insanı
 - <http://evan.quuxuum.org/bgnw.html>
- *Wozniak & Jobs Apple'ı 1977'de kurdu*
 - 1980'de bir garajdan 120 milyon dolarlık satışa ulaştı
- *IBM 1980'de PC'yi tanıttı*
 - Apple Macintosh ile 1984'de yanıt verdi
- *Stroustrup 1980'lerde C++'ı geliştirdi*
 - C dilinin nesneye dayalı uzantısı

İlk kişisel üretkenlik yazılımları – 1978/1979

- Wordstar
- VisiCalc

```
A:HENDOC FC=1 FL=1 COL 01
<<<  MAIN MENU  >>>
--Cursor Movement--  -Delete-  -Miscellaneous-  -Other Menus-
^S char left ^D char right ^G char ^I Tab ^B Reform (from Main only)
^A word left ^F word right ^DEL chr lf ^V INSERT ON/OFF ^J Help ^K Block
^E line up ^X line down ^T word rt ^L Find/Replce again ^Q Quick ^P Print
--Scrolling-- ^Y line ^RETURN End paragraph ^O Onscreen
^W up line ^Z down line ^N Insert a RETURN
^R up screen ^C down screen ^U Stop a command

THIS IS A DOCUMENT BEING WRITTEN ON THE WORDSTAR WORD PROCESSOR ON A KAYPRO
COMPUTER WHICH RUNS UNDER THE CP/M OPERATING SYSTEM.

WORDSTAR WAS A VERY ANKWARD WORD PROCESSOR BY TODAY'S STANDARDS, BUT IN
ITS HEYDAY, IT OFFERED ELECTRONIC WORD PROCESSING TO HUNDREDS OF THOUSANDS
OF PEOPLE WHO WOULD OTHERWISE HAVE NOT BEEN ABLE TO AFFORD IT.

LIKE THE OSBORNE COMPUTER, THE KAYPRO WAS CONSIDERED A "PORTABLE" MACHINE,
ALL 30 POUNDS OF IT. LUGGING ONE OF THESE BEAUTIES AROUND WAS A TASK, AND
SINCE THEY RAN ON AC POWER AND NOT BATTERIES, THEY WERE NOT USABLE EXCEPT
IN A BUILDING OR WHEREVER A POWER SOURCE WAS PRESENT.

LOOKING AT THIS MONOCHROME 8" SCREEN MAY SEEM LUDICROUS BY COMPARISON TO
TODAY'S LAPTOPS, BUT PEOPLE MARVELED AT THIS MACHINE IN THE EARLY 1980s.
```

C11 (L) TOTAL					C1
					25
	A	B	C	D	
1	ITEM	NO.	UNIT	COST	
2	MUCK RAKE	43	12.95	556.85	
3	BUZZ CUT	15	6.75	101.25	
4	TOE TONER	250	49.95	12487.50	
5	EYE SNUFF	2	4.95	9.90	
6					
7			SUBTOTAL	13155.50	
8			9.75% TAX	1282.66	
9			TOTAL	14438.16	

Yazılım Kuşakları

- 1. Jenerasyon:
 - *Programlama fiziksel olarak kablolarla, sonralarda da makine dili ile*
- 2. jenerasyon:
 - *Kullanıcı dostu olmayan ilk kuşak “diller”: assembly*
- 3. jenerasyon:
 - *Yüksek seviyeli diller, sembolik kullanımına gerek kalmadı*
 - *ForTran: 1952 (IBM 704 için), COBOL 1959, BASIC, PASCAL, C, ...*
- 4. jenerasyon:
 - *Genel programlama dili kullanmadan uygulama geliştirme imkanı*
 - *Veritabanı programlama ortamları*
- 5. jenerasyon ??
 - *Kimine göre şu anda kullandığımız IDE’ler!!*
 - *Olması gereken: Doğal dil kullanımı ile sezgisel yaklaşımla geliştirilen programlar*
 - *Kendi kodunu değiştirebilen otonom yazılımlar?*

Programlamanın Evrimi: Makine Dili

- Geç 1940 Erken 1950’lerde programcılar kodu makineye direk olarak veriyordu
- Her makinenin kendi komut seti (0-1 dizisi) vardı
 - *Emek yoğun ve hataya açık*

Programlamanın Evrimi: Assembly dili

■ 1950'lerin ortalarında nümerik kodları hatırlanabilir isimlerle değiştiren assembly dilleri ortaya çıktı

■ Assembler, assembly kodunu makine koduna dönüştüren bir programdır

– *Input: Assembly dilindeki program*

– *Output: Makine dilindeki komutlar dizisi*

■ Hala alt seviyeli ve makineye özel, ancak programlama daha kolay

```
gcc2_compiled.:
    .global _Q_qtod
    .section      ".rodata"
        .align 8
.LLC0:  .asciz   "Hello world!"
    .section      ".text"
        .align 4
    .global main
    .type        main,#function
    .proc        04
main:    !#PROLOGUE# 0
        save %sp,-112,%sp
        !#PROLOGUE# 1
        sethi %hi(cout),%o1
        or %o1,%lo(cout),%o0
        sethi %hi(.LLC0),%o2
        or %o2,%lo(.LLC0),%o1
        call __ls__7ostreamPcc,0
        nop
        mov %o0,%l0
        mov %l0,%o0
        sethi %hi(endl__FR7ostream),%o2
        or %o2,%lo(endl__FR7ostream),%o1
        call __ls__7ostreamPFR7ostream_R7ostream,0
        nop
        mov 0,%i0
        b .LL230
        nop
.LL230: ret
        restore
.LLfe1: .size    main,.LLfe1-main
        .ident   "GCC: (GNU) 2.7.2"
```


İlk Assembler: Initial Orders, 1949

- EDSAC (Electronic Delay Storage Automatic Calculator) için

Order bit pattern	Loc	Order	Meaning	Comment
00101 0 0000000000 0	0:	T0S	$m[0]=A$; $ABC=0$	
10101 0 0000000010 0	1:	H2S	$R=m[2]$	Put $10 \ll 11$ in R
00101 0 0000000000 0	2:	T0S	$m[0]=A$; $ABC=0$	
00011 0 0000000110 0	3:	E6S	goto 6	Jump to main loop
00000 0 0000000001 0	4:	P1S	data 2	The constant 2
00000 0 0000000101 0	5:	P5S	data 10	The constant 10
00101 0 0000000000 0	6:	T0S	$m[0]=A$; $ABC=0$	Start of the main loop
01000 0 0000000000 0	7:	I0S	$m[0]=rdch()$	Get operation code
11100 0 0000000000 0	8:	A0S	$A+=m[0]$	Put it in A
00100 0 0000010000 0	9:	R16S	$ABC \gg = 6$	Shift and store it
00101 0 0000000000 1	10:	T0L	$w[0]=AB$; $ABC=0$	so that it becomes the senior 5 bits of $m[0]$ $m[1]$ is now zero

Programlamanın evrimi: Yüksek seviyeli diller

■ 1950'lerin sonlarından günümüze:

■ Yüksek seviyeli programlama dilleri, programcının daha yüksek bir soyutlama ile düşünmesine olanak tanır.

■ Derleyici, yüksek seviyeli dille yazılmış kodu makine koduna çevirir.

– *Input: c++dilinde program kodu*

– *Output: Makine kodunda komut dizisi*

– *Assembler'a benzer ancak daha karmaşık şekilde*

■ Yorumlayıcı, programda programlama dili ile yazılmış her bir ifadeyi sıralı olarak okuyup işleten aracı programdır

– *Java programları önce sanal Java makinesi koduna derlenir (Java byte code)*

– *Sonra bir yorumlayıcı (Java Sanal Makinesi) ile işletilir.*

```
/**
 * This class can print "Hello world!"
 * @author Dave Reed
 * @version 8/20/04
 */

class Greeter
{
    public Greeter() { }

    public void SayHello() {
        System.out.println("Hello world!");
    }
}
```

İlk yüksek seviyeli dil: Plankalkül (1948)

- Konrad Zuse, Z2-Z3 için
- 30 yıl sonra, Joachim Hohmann tezinde Plankalkül için derleyiciyi yarattı.

Die Punkte sind benachbart

	$V \neq V$	$V \wedge$	$ V - V \leq L$	$ V - V \leq L \Rightarrow R \Delta . 17$
V	0	1	0	1
K			0	0
			1	1

İlk Derleyici: A-0 (1952)

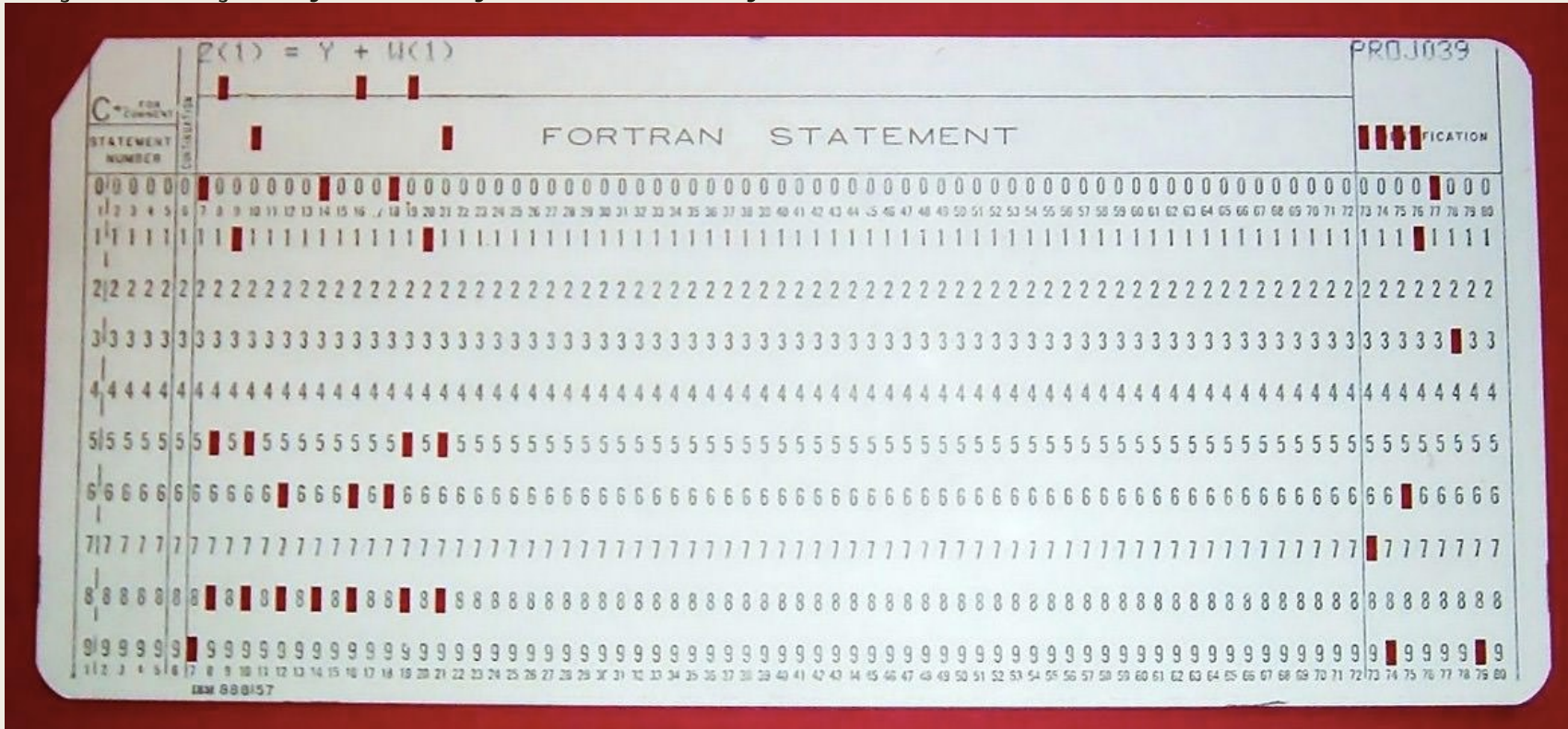
İlk açık kaynak kodlu yazılım:A-2

- UNIVAC için, Grace Hopper tarafından
- A-0, sonra A-2'ye dönüştü.
- Müşterilere kaynak kodu ile birlikte veriliyordu
 - *İlk açık kaynak kodlu yazılım!*



İlk Yaygın kullanılan yüksek seviyeli Programlama Dili: Fortran

- 1963'te 40'dan fazla FORTRAN derleyicisi vardı.
- Her işlemci için ayrı derleyici üretilmeliydi.



İlk Nesneye Dayalı Dil: Simula (1967)

```
Begin
  Class Glyph;
    Virtual: Procedure print Is Procedure print;
  Begin
  End;
  Glyph Class Char (c);
    Character c;
  Begin
    Procedure print;
      OutChar(c);
    End;
  Ref (Glyph) rg;
  rg := New Char ('A');
  rg.print;
End;
```

Mutlak adresleme

- 1010010101110101011

Panel	1	function: enter house
Switch 0	1	open the door
Switch 1	1	put the lights on
Switch 2	0	close the door (please)

Makine Dili

Label	Opcode	Register
CALC:	STO	R1, HELP0
	STO	R2, HELP2
	LD	R3, HELP1
	ADD	R3, HELP2
	LD	R4, HELP1
	SUB	R4, HELP2
	RSR	SP, 0
HELP1:	DS	2
HELP2:	DS	2

Alt programlar

Start of program
the main "menu"

first subroutine

back to the main menu
second subroutine
*with a parameter (contents of
what to print)*

back to procedure: main

Begin program;

Main;

Printf ("Hello World");

DoSomethingElse()

Printf ("Hello World");

(end of program)

Function **DoSomethingElse;**

Add two numbers;

Return OK

Function **Printf**(what_to_print)

Open channel to printer interface;

Initialize printer;

Send "what_to_print" to printer;

Send page feed to printer;

Close printer interface;

Return OK

Yüksek Seviyeli Programlama Dilleri

human	computer
Add 2 and 2	<code>answer := 2+2;</code>
Show me the answer	<code>printf ("%d\n", answer);</code>

Yapay Zeka Dilleri

- LISP 1958

```
;;; HWorld.lsp
```

```
(DEFUN HELLO ()  
  "HELLO WORLD"  
)
```

- Prolog 1970

```
Hello:-  
  nl,  
  write('Hello world!' ).  
}
```

- Smalltalk 1979

- Algol 1960

- Simula 1967

Nesneye Dayalı Diller

- İlk nesne dili :SIMULA ~1970
- İlk nesneye dayalı dil: SmallTalk 1979
- C++: 1980'lerin ortaları
- JAVA: 1990'lar
- C#: 2000'ler
- OOP-Procedural hibrit diller: Python, Ruby, ...

Gelecek?

- Yapay Zeka
- Kendini Değiştiren nesneler
- Uygulama yaratma arayüzleri
- ...

Gelecek Ders

- Olasılıksal Robotik Çalışma Grubu Sunumu