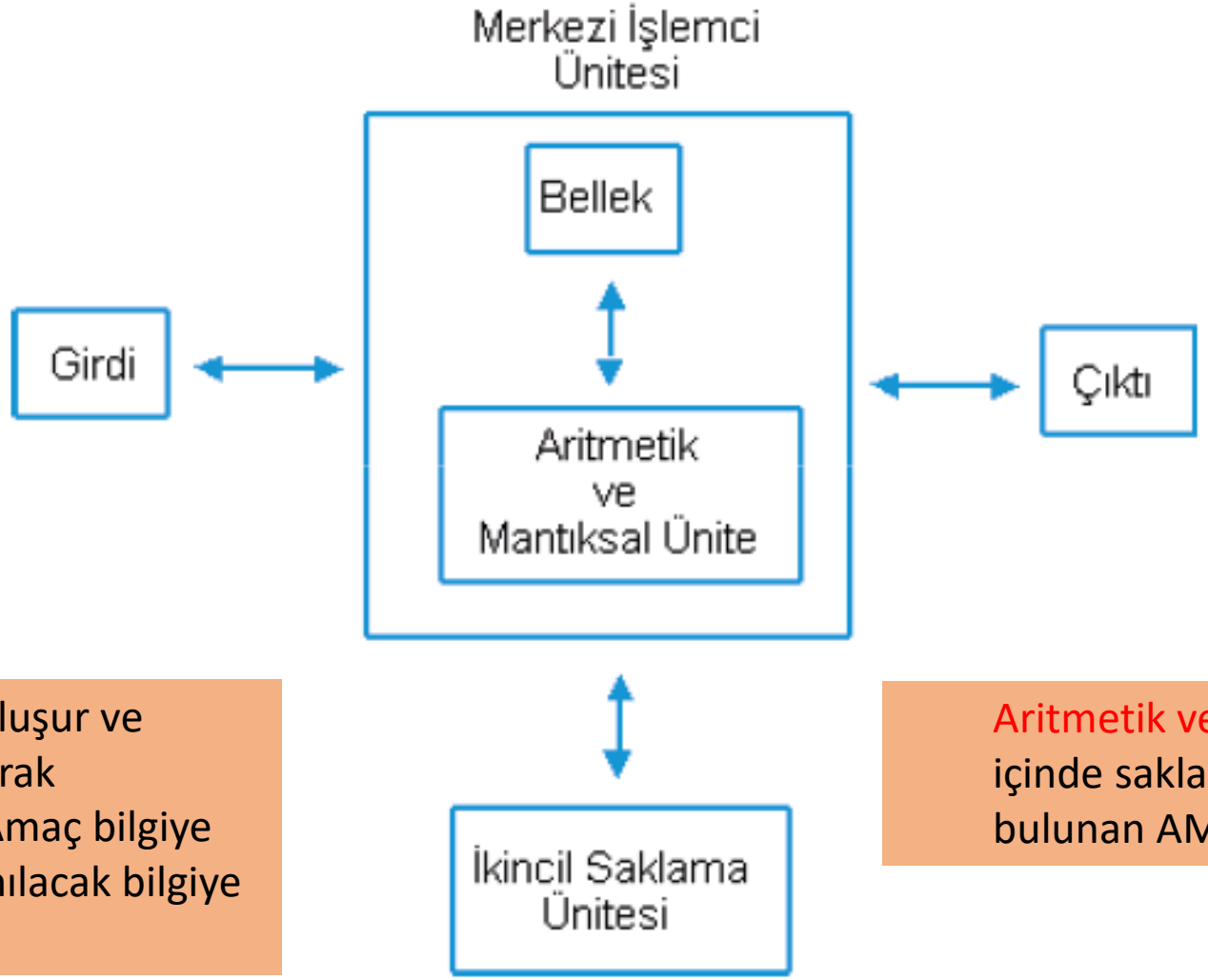
UYB103
Programlama I

Bölüm 1- Genel Kavramlar

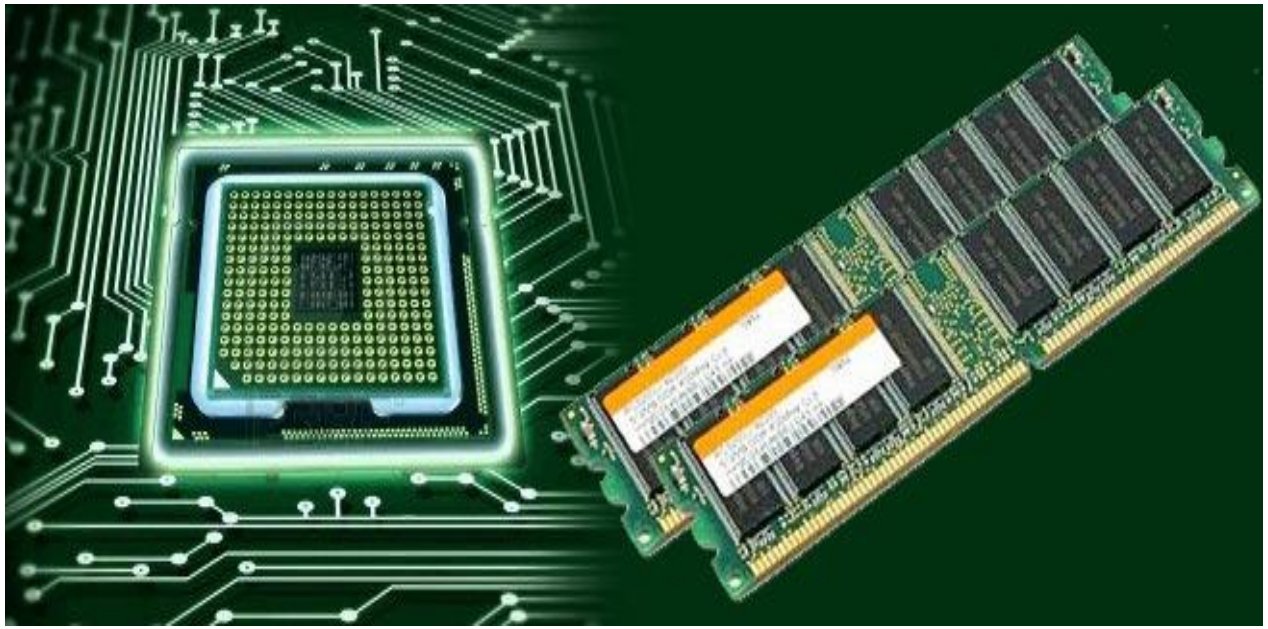
Bilgisayarın Temel Birimleri



Bellek: hücrelerden oluşur ve bilginin kısa süreli olarak saklanmasını sağlar. Amaç bilgiye kısa süre içinde kullanılacak bilgiye hızlı erişmektir.

Aritmetik ve Mantıksal Ünite: Bellek içinde saklanmış olan bilgi MİÜ içinde bulunan AMÜ yardımıyla işenir

MİÜ - RAM

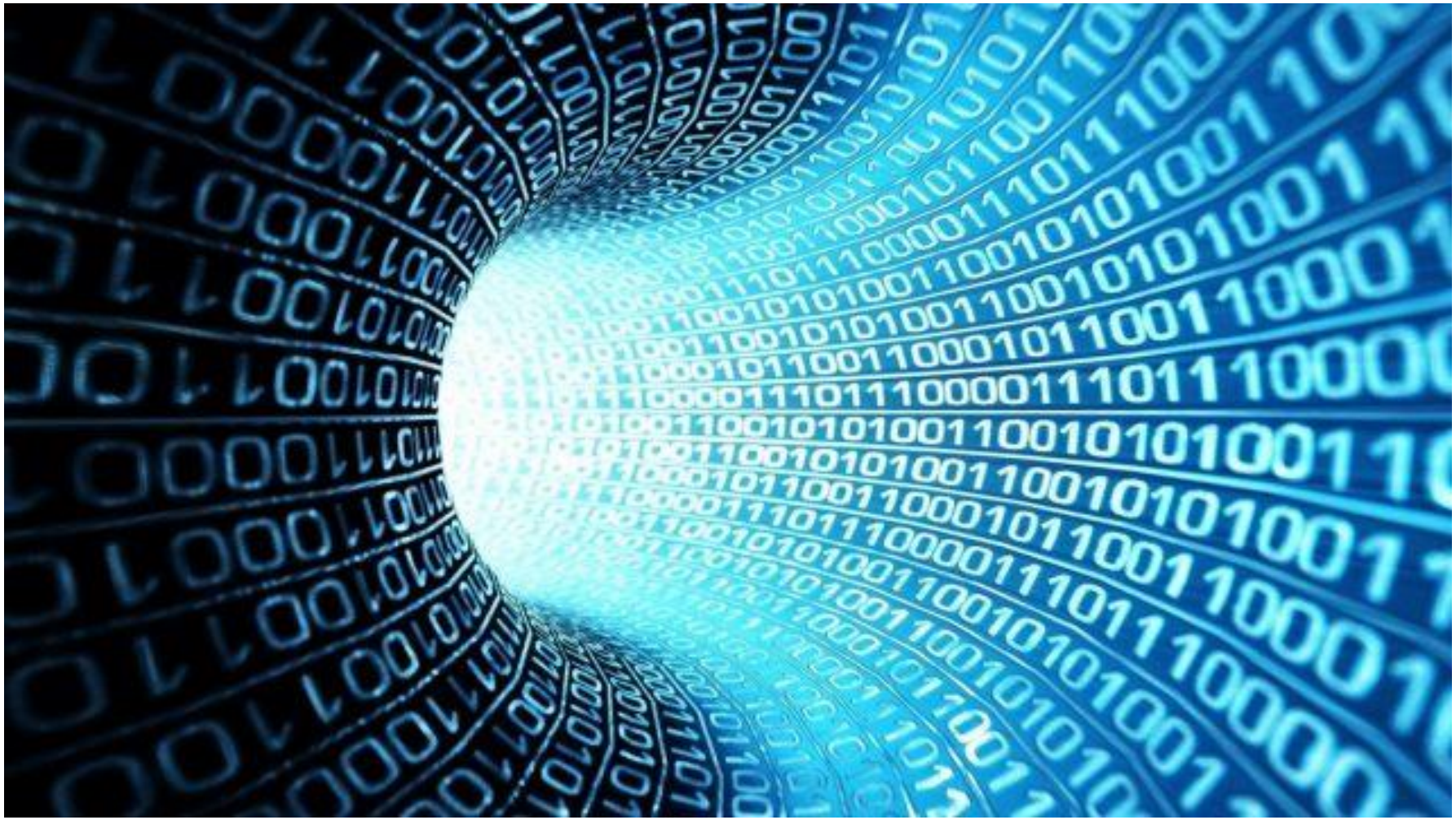


İkincil Saklama Ünitesi

Bilgisayarın Temel Birimleri

- Giriş Üniteleri: bilgisayara veri girişini sağlayan bölümlerdir (fare, klavye, tarayıcılar)
- Çıkış Üniteleri: bilgisayarımızdan veri çıkışı yaptığımız bölümlerdir
- Merkezi İşlemci Ünitesi (MİÜ):
  - Bellek: hücrelerden oluşur ve bilginin kısa süreli olarak saklanmasını sağlar. Amaç bilgiye kısa süre içinde kullanılacak bilgiye hızlı erişmektir.
  - Aritmetik ve Mantıksal Ünite: Bellek içinde saklanmış olan bilgi MİÜ içinde bulunan AMÜ yardımıyla işenir
- İkincil Saklama: bilgi daha uzun süreli olarak saklanmak istenirse bu amaçla ikincil saklama üniteleri kullanılır (disk, teyp, CD, flash disk)

Bilgisayarın Alfabesi



Bilgisayarın Alfabesi
(Binary System)

- Bizler iletişim kurmak için 29 harf ve 10 rakamdan oluşan bir sistem kullanırız.
- Tüm sözlü iletişimimiz bu karakterlerin çeşitli varyasyonlarda oluşturduğu formlarda olur.

- Bilgisayarlar da benzer şekilde kendi alfabelerini kullanırlar.
- Bilgisayarların kullandığı alfabeye, makine dili (machine language) adı verilir.
- Bu alfabe sadece 0 ve 1 rakamlarından oluşur

01100
10110
11110

Bilgi Saklama

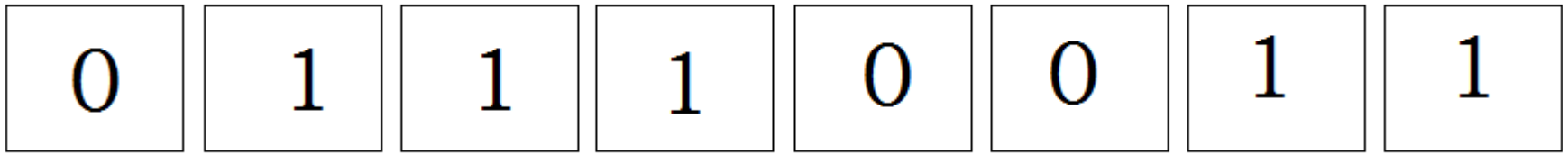
- Bilgisayarda bilgi saklamaya yarayan en küçük elektronik bilgi alanı **bit** olarak adlandırılır.
- Bir **bit** içinde 1 ya da 0 değeri taşıyabilen elektronik bilgi saklama alanıdır.

0	Sıfır değeri içeren bir bit
1	Bir değeri içeren bir bit

- Bir bit içinde saklanan 1 ya da 0 değeri;
  - ✓ Bir devrenin açık ya da kapalı olmasını
  - ✓ Bir ifadenin doğru ya da yanlış olmasını
  - ✓ Bir cevabın evet ya da hayır olmasını tanımlamak için kullanılabilmektedir

Bayt (Byte)

- Veri saklarken, 8 bitin yanyana getirilerek kullanımı çok yaygın bir hale gelmiştir ve bu nedenle sekiz bitlik veri gruplarına **bayt (byte)** adı verilmiştir.
- Bir baytlık bilgi alanı, içindeki her bir bitlik bilgi alanına farklı değerler atayarak, değişik bilgileri saklayabilir.
- Bir bayt alanı içinde 256 (2^8) farklı bilginin (durumun) saklanması mümkündür.
- Böylece karakterlerin ve sayıların 1 bayt bilgi alanı içinde tanımlanması mümkün.



2lik Sayı Sistemi

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
	1	1	1	1	1	0	1

n	Bölüm	Kalan
125	62	1
62	31	0
31	15	1
15	7	1
7	3	1
3	1	1
1	0	1

10luk Sayı Sistemi

10^3	10^2	10^1	10^0
1000	100	10	1
	1	2	5

n	Bölüm	Kalan
125	12	5
12	1	2
1	0	1

2lik Sayı Sisteminden 10luk Sisteme Geçiş

$$(125)_{10}=(1111101)_2$$

$$125 = (1*2)^0+(0*2)^1+(1*2)^2+(1*2)^3+(1*2)^4+(1*2)^5+(1*2)^6$$

Bellek Kapasitesi Dönüştürme Oranı

kapasite	sembol		deger
1 bit	Bit	=	0 **veya** 1
1 byte	Byte	=	8 bit
1 Kilobyte	KB	=	1024 bytes
1 Megabyte	MB	=	1024 KB
1 Gigabyte	GB	=	1024 MB
1 Terabyte	TB	=	1024 GB
1 Petabyte	PB	=	1.024 TB

Alfasayısal Değerler ve Özel Karakterler

- **Alfasayısal değerler:** karakterlerin ve sayısal değerlerin tamamını kapsayan değerler kümesidir.
  - 1,2 gibi sayısal değerleri içerebileceği gibi, A,B gibi alfabetik karakterleri de kapsar

- Alfasayısal değerlerinin ve noktalama işaretleri gibi diğer özel karakterlerin bilgisayar sisteminde saklanması ve kullanılması amacıyla her bir değere **karşılık gelen bir sayısal değer atanmıştır**.

- Bu değerler ASCII (American Standard Code for Information Exchange) kod tablosunda tanımlanmıştır.

ASCII Kod Tablosu

	0	1	2	3	4	5	6	7	8	9
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dcl	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	b	!	"	#	$	%	&	‘
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	del		

Algoritmalar

- Bilgisayara bir işin nasıl yapılması gerektiği ile ilgili tüm ayrıntıların detaylı ve düzenli bir biçimde aktarılması gerekir.

- Bilgisayarlara bir işin nasıl yapılması gerektiğini **algoritmalar (algorithms)** ile tanımlarız.

- Algoritmalar yaptırılmak istenilen işin hangi adımları takip edilerek gerçekleştirilmesi gerektiğini anlatan adımlardır.

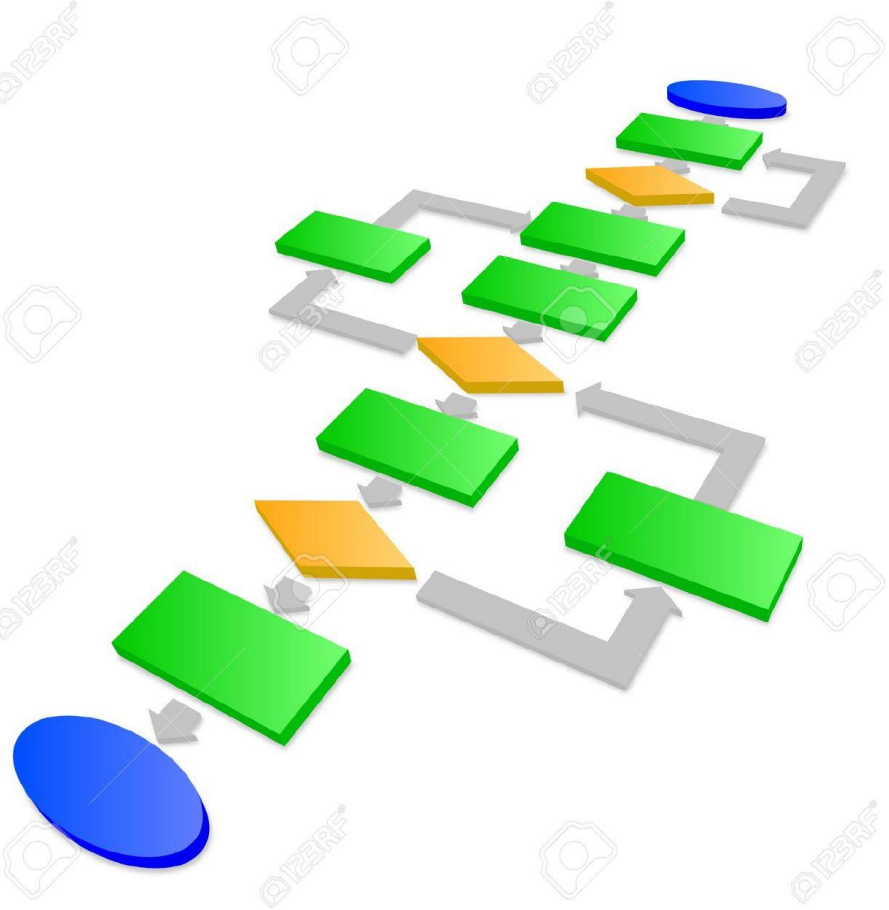
- Bu adımlar daha sonra bilgisayarın anlayabileceği program komutlarına dönüştürülecektir

- Bir algoritmayı **akış şeması (flow charts**) yardımı ile görsel olarak düzenleyebiliriz.

Akış Şeması

- Akış şemaları: algoritmalarda verilen her adımın görsel olarak anlatılması amacıyla kullanılan yöntemlerden biridir.

- Akış şemaları ile, bilgisayarın çözmesini istediğimiz problem ile ilgili, olabilecek bütün adımları tanımlarız

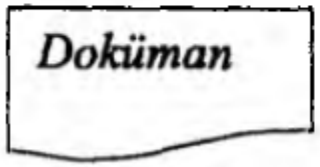


Akış Şeması

- ***İşlem*** (process): Yapılması istenen işlemle ilgili tanımlamalar dikdörtgen şeklindeki kutucuklar yardımıyla gösterilir. İşlemlerle ilgili komutlar kutu içine yazılır.
- ***Karar*** (decision): Bazı kararlara bağlı olarak farklı yolların izlenebileceği durumlarda baklava şeklindeki bu gösterim kullanılır. Bu şekil içinde tanımlanan koşula göre farklı kararların alınması sağlanır.

Akış Şeması

- ***Veri*** (data): Tanımlanan işlemler bazı verileri kullanıyorsa ve bu verilerin farklı bir ortamda tutulması gerekiyorsa bu şekil kullanılır. Her farklı veri grubu için farklı isim verilmelidir.



- ***Doküman:*** Program sırasında çıktı olarak gösterilmek istenen veriler ve işlemler bu şekil ile tanımlanır.

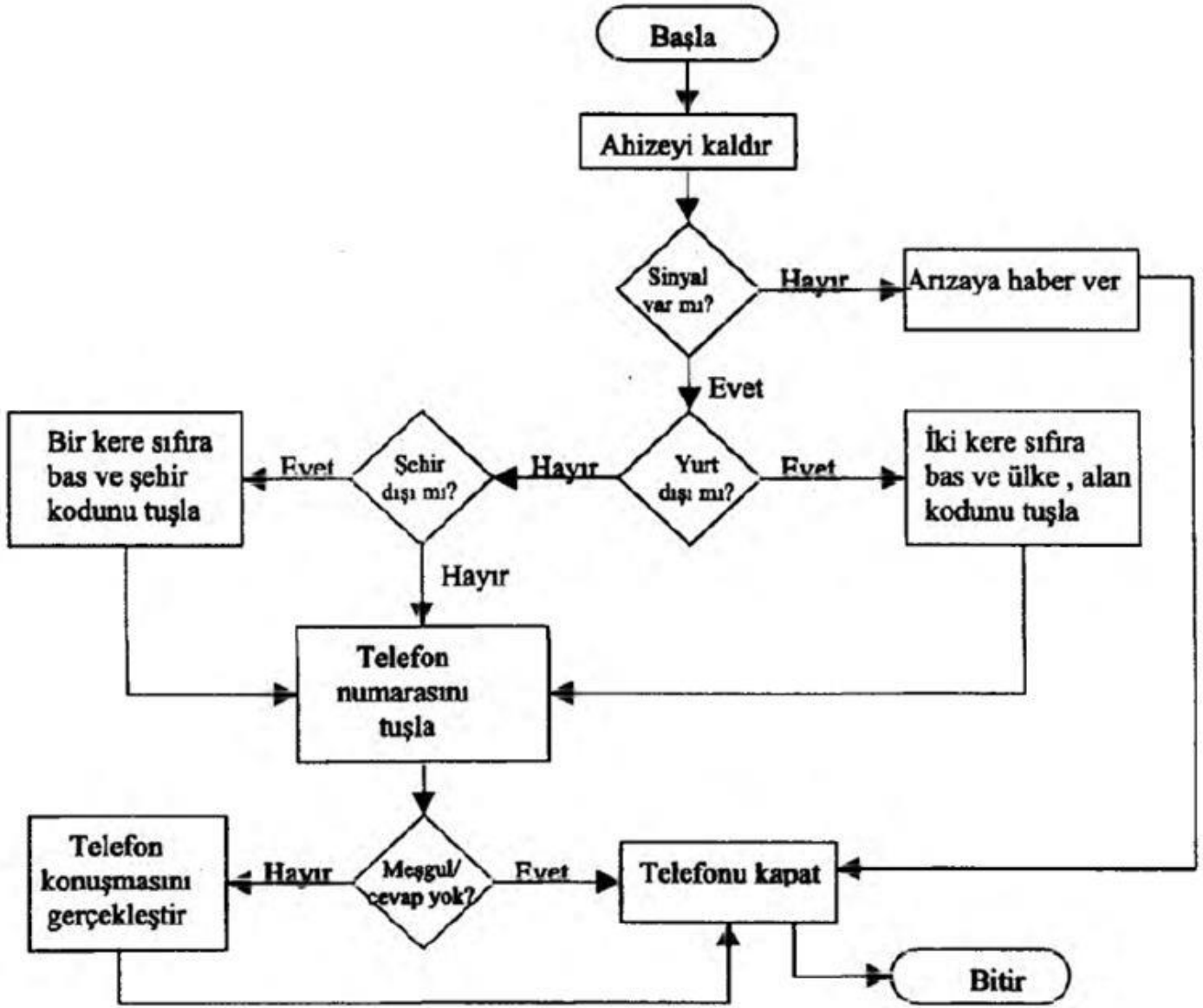
- ***Bağlantı:*** Genellikle büyük yazılımlar için hazırlanan akış şemaları bir sayfada görüntülenemez. Anlaşılabilirliği artırmak ve bir sonraki sayfada işlemlerin hangi noktadan devam edeceğini göstermek amacıyla bağlantılar kullanılır.

- ***Başla/Bitir:*** Akış şemasının komutlarının başlangıç ve bitiş noktaları bu gösterim kullanılarak belirlenir.

Telefon Etme Algoritması ?

1. Başla
2. Ahizeyi kaldır
3. Sinyali kontrol et, sinyal yoksa arızaya haber ver ve 9. adım'a git
4. Eğer telefon numarası yurt dışında ise iki kere sıfır tuşuna bas ve ülke ve alan kodunu tuşla ve 6. adıma git
5. Eğer telefon numarası şehir dışında ise bir kere sıfır tuşuna bas ve alan kodunu tuşla
6. Telefon numarasını tuşla
7. Eğer hat meşgul ise ya da cevap vermiyorsa 9.adıma git
8. Telefon konuşmasını gerçekleştir
9. Telefonu kapat
10. Bitiş

Telefon Etme Algoritması Akış Şeması



Neden C?


/* This line basically imports the "stdio" header file, part of
* the standard library. It provides input and output functionality
* to the program.
*/
#include <stdio.h>
/*
* Function (metho ion. This outputs "Hello, world" to
* standard output ked.
*/
void sayHello() {
// printf() in he specifi d text (with optional
// formatting o
printf("Hello, w
}
/*
* This is a "main function". The compiled program will run the code
* defined here.

Neden C?

- Neden C sorusuna cevap olabilecek birkaç madde:
  1. C dili ona harcadığınız her türlü emeğe değecek kadar güzel ve güçlü bir dildir
  2. C dili bir klasiktir. Bugün kullandığımız birçok teknoloji doğrudan ya da dolaylı olarak C temellidir. Özellikle işletim sistemi, sürücü ve derleyici gibi sistem yazılımları için C (C++) dili vazgeçilmezdir.
  3. C dili ile programlama temellerini çok iyi öğrenebilirsiniz.
  4. C öğrenmekle, C++ öğrenmek için çok güzel ve çok büyük bir adım atmış olursunuz.
  5. C dilini öğrendikten sonra diğer dilleri kavramanız daha kolay olacaktır.

Programlama Dilleri ve Seviyeleri

- Programlama dilleri seviyelerine göre üç kısıma ayırabiliriz:

1. **Düşük seviyeli diller (Makine dili, Assembly dili)**
2. **Orta seviyeli diller (C)**
3. **Yüksek seviyeli diller (Visual Basic, MS Access)**

NOT:

Bir programlama dilinin seviyesi, o programlama dilinin ne kadar iyi bir programlama dili olduğunun bir göstergesi değildir.

Bir dilin seviyesi o dilin makine diline olan yakınlığının bir ölçüsüdür

Bir dil makine diline ne kadar yakınsa o kadar düşük seviyeli bir dildir

Bir dil makine diline ne kadar uzaksa o kadar yüksek seviyeli bir dildir

Makine Dilinde (Merhaba Dünya kodu)

01001000 01100101 01101100 01101100
01101111 00100000 01010111 01101111
01110010 01101100 01100100 00100001
00100000

Assembly Dili (Merhaba Dünya kodu)

```
01 DATA SEGMENT
02       MESSAGE DB "HELLO WORLD!!!$"
03 ENDS
04
05 CODE SEGMENT
06      ASSUME DS:DATA CS:CODE
07 START:
08         MOV AX,DATA
09         MOV DS,AX
10         LEA DX,MESSAGE
11         MOV AH,9
12         INT 21H
13         MOV AH,4CH
14         INT 21H
15 ENDS
16 END START
17
```

C Dilinde (Merhaba Dünya kodu)

```
#include <stido.h>
int main()
{
        printf (“Merhaba Dunya”);
        return (0);
}
```

Editörler

- Editörler, kodlarımızı yazmak için kullandığımız programlara verdiğimiz genel isimdir.

- Editörler kullanarak aşağıda listelenen eylemleri yapabiliriz:
  - Kodlarımızı düzenleriz
  - Kodlarımızı kaydederiz
  - Kodlarımızı derleriz
  - Kodlarımızı çalıştırarak hatalarını ayıklarız

- Editörler bize daha hızlı, etkili ve güvenli kod yazmamızı sağlamaktır.

Dev-C++

İndirme linki: https://sourceforge.net/projects/orwelldevcpp/?source=typ_redirect