

# Sistem Programlama

## Ders 3

Dr. Öğr. Üyesi Mehmet Dinçer Erbaş  
Abant İzzet Baysal Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü

# Unix standartları

- Unix uygulamaları mümkün olduğunca farklı Unix sistemlerinde çalışabilecek şekilde dizayn edilmiştir.
- Ancak 1980'ler ile birlikte birçok Unix sistemi oluşmuş ve uygulamalarda farklılıklar ortaya çıkmıştır.
- Bu durum Unix sisteminin ortaya çıkış nedenlerinden biri olan farklı sistemlerde çalışabilme özelliğine aykırıdır.
- Bu sebeple birçok kullanıcı sistemin standartlarının belirlenmesi konusunda istekte bulunmuştur.

# Unix standartları

- 1989 senesinde C dilinin ANSI standartları kabul edilmiştir.
- Bu standartlar ayrıca ISO ve IEC tarafından 1990 yılında kabul edilmiştir.
  - ANSI: American National Standards Institute
  - ISO: International Organization for Standardization
  - IEC: International Electrotechnical Commission
- C standartları günümüzde ISO/IEC tarafından takip edilmektedir.
  - Bu gurpların amacı C dilinde yazılan programların birçok farklı işletim sisteminde düzgün şekilde çalışmasına olanak verecek standartların geliştirilmesidir.
- Bu standartlar sadece dile ait yazım ve anlam kurallarını belirlemez, ayrıca standart kütüphaneleri ve bu kütüphanelerde bulunması gereken fonksiyonları belirtir.
- ISO C kütüphanesi tanımlanan header dosyaları ile 24 farklı alana ayrılabilir.
  - Bakınız: ISO C headers.

# Unix standartları

Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
<assert.h>	•	•	•	•	verify program assertion
<complex.h>	•	•	•	•	complex arithmetic support
<ctype.h>	•	•	•	•	character classification and mapping support
<errno.h>	•	•	•	•	error codes (Section 1.7)
<fenv.h>	•	•	•	•	floating-point environment
<float.h>	•	•	•	•	floating-point constants and characteristics
<inttypes.h>	•	•	•	•	integer type format conversion
<iso646.h>	•	•	•	•	macros for assignment, relational, and unary operators
<limits.h>	•	•	•	•	implementation constants (Section 2.5)
<locale.h>	•	•	•	•	locale categories and related definitions
<math.h>	•	•	•	•	mathematical function and type declarations and constants
<setjmp.h>	•	•	•	•	nonlocal goto (Section 7.10)
<signal.h>	•	•	•	•	signals (Chapter 10)
<stdarg.h>	•	•	•	•	variable argument lists
<stdbool.h>	•	•	•	•	Boolean type and values
<stddef.h>	•	•	•	•	standard definitions
<stdint.h>	•	•	•	•	integer types
<stdio.h>	•	•	•	•	standard I/O library (Chapter 5)
<stdlib.h>	•	•	•	•	utility functions
<string.h>	•	•	•	•	string operations
<tgmath.h>	•	•	•	•	type-generic math macros
<time.h>	•	•	•	•	time and date (Section 6.10)
<wchar.h>	•	•	•	•	extended multibyte and wide character support
<wctype.h>	•	•	•	•	wide character classification and mapping support

# IEEE Posix

- Posix standartları IEEE tarafından geliştirilmiştir.
  - IEEE: Institute of Electrical and Electronics Engineers
  - Posix: Portable Operating System Interface
- 1003.1 İşletim sistemi arayüz standartları bir işletim sisteminin diğer programlara sağlaması gereken olanakları tanımlar.
  - Bu standartlar çoğu bilgisayar üreticisinde kabul edilmiştir.
  - Eğer bu olanaklar sağlanıyorsa, belirtilen işletim sistemi POSIX uyumludur.
  - Bu standartlar sadece Unix veya Unix türevleri ile sınırlı değildir.
  - Bazı üreticiler kendi işletim sistemlerini Posix uyumlu hale getirmiştir.
- 1003.1 standartları bir arayüz tanımlar, bu arayüzün arka planında çalışan işlemleri tanımlamaz.

# Unix standartları

Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
<aio.h>	•	•	•	•	asynchronous I/O
<cpio.h>	•	•	•	•	cpio archive values
<dirent.h>	•	•	•	•	directory entries (Section 4.22)
<dlfcn.h>	•	•	•	•	dynamic linking
<fcntl.h>	•	•	•	•	file control (Section 3.14)
<fnmatch.h>	•	•	•	•	filename-matching types
<glob.h>	•	•	•	•	pathname pattern-matching and generation
<grp.h>	•	•	•	•	group file (Section 6.4)
<iconv.h>	•	•	•	•	codeset conversion utility
<langinfo.h>	•	•	•	•	language information constants
<monetary.h>	•	•	•	•	monetary types and functions
<netdb.h>	•	•	•	•	network database operations
<nl_types.h>	•	•	•	•	message catalogs
<poll.h>	•	•	•	•	poll function (Section 14.4.2)
<pthread.h>	•	•	•	•	threads (Chapters 11 and 12)
<pwd.h>	•	•	•	•	password file (Section 6.2)
<regex.h>	•	•	•	•	regular expressions
<sched.h>	•	•	•	•	execution scheduling
<semaphore.h>	•	•	•	•	semaphores
<strings.h>	•	•	•	•	string operations
<tar.h>	•	•	•	•	tar archive values
<termios.h>	•	•	•	•	terminal I/O (Chapter 18)
<unistd.h>	•	•	•	•	symbolic constants
<wordexp.h>	•	•	•	•	word-expansion definitions

# Unix standartları

Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
<code>&lt;arpa/inet.h&gt;</code>	•	•	•	•	Internet definitions (Chapter 16)
<code>&lt;net/if.h&gt;</code>	•	•	•	•	socket local interfaces (Chapter 16)
<code>&lt;netinet/in.h&gt;</code>	•	•	•	•	Internet address family (Section 16.3)
<code>&lt;netinet/tcp.h&gt;</code>	•	•	•	•	Transmission Control Protocol definitions
<code>&lt;sys/mman.h&gt;</code>	•	•	•	•	memory management declarations
<code>&lt;sys/select.h&gt;</code>	•	•	•	•	<code>select</code> function (Section 14.4.1)
<code>&lt;sys/socket.h&gt;</code>	•	•	•	•	sockets interface (Chapter 16)
<code>&lt;sys/stat.h&gt;</code>	•	•	•	•	file status (Chapter 4)
<code>&lt;sys/statvfs.h&gt;</code>	•	•	•	•	file system information
<code>&lt;sys/times.h&gt;</code>	•	•	•	•	process times (Section 8.17)
<code>&lt;sys/types.h&gt;</code>	•	•	•	•	primitive system data types (Section 2.8)
<code>&lt;sys/un.h&gt;</code>	•	•	•	•	UNIX domain socket definitions (Section 17.2)
<code>&lt;sys/utsname.h&gt;</code>	•	•	•	•	system name (Section 6.9)
<code>&lt;sys/wait.h&gt;</code>	•	•	•	•	process control (Section 8.6)

Single Unix spesifikasyonu opsiyonel header dosyaları ile Posix standartlarını genişletir.

# Unix standartları

- Limitler
  - Unix sistemlerinde birçok özel sayı ve sabit tanımlanmıştır
  - İki farklı limit tipine ihtiyaç vardır. Bunlar:
    - Derleme zamanı limitleri
      - Örneğin bir short int değişkenin alabileceği maksimum değer.
    - Çalışma zamanı limitleri
      - Örneğin bir dosya isminde en fazla kaç karakter olabilir.
  - Derleme zamanı limitleri header dosyalarında saklanır ve programlar bu dosyaları ekleyerek bu limitleri kullanabilir.
  - Çalışma zamanı limitlerine erişebilmek için işlemler bir fonksiyon çağırırlar ve limit değerine erişirler.



# Unix standartları

- Limitlerle ilgili aşağıda belirtilen problemler gözlemlenebilir.
  - Bazı limitler bir sistemde sabit olarak oluşturulmuş ve header dosyasında bulunurken, başka bir sistemde farklı değere sahip ve bir çalışma zamanı fonksiyonu ile değeri öğreniliyor olabilir.
    - Örneğin SVR4 öncesinde, System V sadece 14 karakterlik dosya isimlerine izin verirken, BSD-tabanlı sistemler 255 karaktere kadar dosya isimlerine izin vermektedir.
    - Günümüzde kullanılan çoğu Unix sisteminde birden fazla dosya sistemi tipi desteklenmekte ve her sistemin kendine ait limitleri bulunmaktadır. Bu sebeple çalışma sırasında belli limitler dosyanın hangi sistemin parçası olduğuna göre değişebilmektedir.

# Unix standartları

- Limitler
  - Bu problemleri çözmek için, üç farklı limit tipi tanımlanmıştır.
    - Derleme zamanı limitleri (header dosyalarında)
    - Dosya veya klasör ile ilgili olmayan çalışma zamanı limitleri (sysconf fonksiyonu)
    - Dosya veya klasör ile ilgili çalışma zamanı limitleri (pathconf ve fpathconf fonksiyonları).

# Unix standartları

- sysconf, pathconf, and fpathconf fonksiyonları
- `long sysconf (int name)`
  - bir dosya veya klasör ile ilgili olmayan çalışma zamanı limitlerini öğrenmek için kullanılır.
- `long pathconf (const char *pathname, int name)`  
ve
- `long fpathconf (int filedes, int name)`
  - Bir dosya veya klasör ile ilgili çalışma zamanı limitlerini öğrenmek için kullanılır

# Unix standartları

- ISO C limitleri
  - ISO C tarafından tanımlanan bütün limitler derleme-zamanı limitleridir.
  - Bir sonraki slaytta <limits.h> dosyasında tanımlanan sabitler görülmektedir.
    - Üçüncü sütunda ISO C tarafından kabul edilen minimum değerler görülmektedir.
      - Bu değere göre 16-bit tam sayılarla birin tamamlayıcısı (one's complement) işlemlerin yapılmasına olanak verir.
    - Dördüncü sütun 32-bit tam sayılarla ikinin tamamlayıcısı (two's complement) işlemleri yapan Linux sistemleri için değerlerdir.

# Unix standartlari

Name	Description	Minimum acceptable value	Typical value
CHAR_BIT	bits in a char	8	8
CHAR_MAX	max value of char	(see later)	127
CHAR_MIN	min value of char	(see later)	-128
SCHAR_MAX	max value of signed char	127	127
SCHAR_MIN	min value of signed char	-127	-128
UCHAR_MAX	max value of unsigned char	255	255
INT_MAX	max value of int	32,767	2,147,483,647
INT_MIN	min value of int	-32,767	-2,147,483,648
UINT_MAX	max value of unsigned int	65,535	4,294,967,295
SHRT_MAX	max value of short	32,767	32,767
SHRT_MIN	min value of short	-32,767	-32,768
USHRT_MAX	max value of unsigned short	65,535	65,535
LONG_MAX	max value of long	2,147,483,647	2,147,483,647
LONG_MIN	min value of long	-2,147,483,647	-2,147,483,648
ULONG_MAX	max value of unsigned long	4,294,967,295	4,294,967,295
LLONG_MAX	max value of long long	9,223,372,036,854,775,807	9,223,372,036,854,775,807
LLONG_MIN	min value of long long	-9,223,372,036,854,775,807	-9,223,372,036,854,775,808
ULLONG_MAX	max value of unsigned long long	18,446,744,073,709,551,615	18,446,744,073,709,551,615
MB_LEN_MAX	max number of bytes in a multibyte character constant	1	6