MŰEGYETEM 1782

A programozás alapjai 1

4. előadás

BME

Híradástechnikai Tanszék

A C nyelv típusai

C típusok

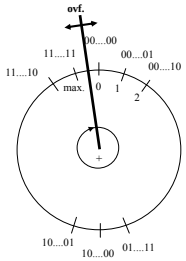
- void
- skalár:
  - aritmetikai:
    - egész:
      - integer
      - karakter
      - felsorolás
    - lebegőpontos
  - mutató
- függvény
- union
- összetett:
  - tömb
  - struktúra

Egész típusok

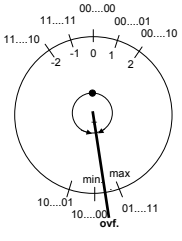
A szabvány nem írja elő, de a gyakorlatban

- előjeles egészeket
  kettes komplemens kódban
- előjel nélküli egészeket
  bináris alakban ábrázolunk

Előjel nélküli egészek bináris ábrázolása



Előjeles egészek kettes komplemens kódú ábrázolása



Az ábrázolás korlátai

Az adott implementációhoz az értékkészletet a **<limits.h>** file adja meg.

Előjeles integer típusok

	minimális hossz [bit]	típus-megadás alakja	minimális érték <limits.h> -ban	maximális érték <limits.h> -ban
rövid	16	short short int signed short signed short int	SHRT_MIN	SHRT_MAX
normál	16	int signed int	INT_MIN	INT_MAX
hosszú	32	long long int signed long signed long int	LONG_MIN	LONG_MAX

Előjeles integer típusok

Szám-konstans alakja:

Típus	Számrendszer	Példák
int	decimális	**0 123 -33**
	oktális	**012 0177777**
	hexadeimális	**0x1a 0x7fff 0xAa1bB**
long		**-"-L -"-l** vagy, ha az érték olyan nagy

Előjel nélküli integer típusok

	minimális hossz [bit]	típus-megadás alakja	minimális érték	maximális érték <limits.h> -ban
rövid	16	unsigned short unsigned short int	0	USHRT_MAX
normál	16	unsigned unsigned int	0	UINT_MAX
hosszú	32	unsigned long unsigned long int	0	ULONG_MAX

Előjel nélküli integer típusok

Szám-konstans alakja:

Típus	Példák
unsigned	<mintha int lenne>**u** <mintha int lenne>**U**
unsigned long	<mintha int lenne>**lu** <mintha int lenne>**Ul** <mintha int lenne>**Lu** <mintha int lenne>**UL** vagy, ha az érték olyan nagy

Automatikus ábrázolási mód

ha nincs megadva U vagy L

Amelyikbe a szám-konstans előbb belefér:

1. **int**
2. **unsigned int** oktális és hexa esetén
3. **long int**
4. **unsigned long int**

A …**short** típusok csak az adat tárolási hosszát írják elő.

Számításnál a ... **short** típusú értéket automatikusan ... **int** típusúvá alakítja, és azzal számol.

Karakter típusok

Ábrázolás: minimum 8 biten.

Ez lesz a tárolás alapegysége

típus-megadás alakja	minimális érték <limits.h> -ban	maximális érték <limits.h> -ban
signed char	SCHAR_MIN	SCHAR_MAX
unsigned char	0	UCHAR_MAX
char	CHAR_MIN	CHAR_MAX

Karakter típusok

Karakter-konstans alakja:

Egyszerű karakterek

Alakja	Jelentése
'a'	kis a betű
'A'	nagy A betű
':'	kettőspont

Karakter típusok

Karakter-konstans alakja:

Különleges karakterek

Alakja	Jelentése
'\''	aposztróf
'\"'	idézőjel
'\\'	backslash
'\?'	kérdőjel
'\a'	hangjelzés

Karakter típusok

Karakter-konstans alakja:

Különleges karakterek

Alakja	Jelentése
'\n'	új sor
'\f'	lapdobás
'\t'	tabulátor
'\v'	függőleges tabulátor
'\b'	visszatörlés

Karakter típusok

Karakter-konstans alakja:

Karakter megadása kódjával

Alakja	Jelentése
'\0'	nullás kódú karakter
'\10'	oktális szám
'\x10'	hexadecimális szám

A …**char** típusok gyakorlatilag
az egészekhez hasonlóan viselkednek.

Számításnál a ... **char** típusú értéket
automatikusan ... **int** típusúvá alakítja,
és azzal számol.

Lebegőpontos típusok

Típus	Konstans alakja	Min. abs. érték	Max. abs. érték	Pontosság [dec. jegy]
float	12.3f 0.12F 12.F .5f 1E-3f 1.8e5f	FLT_MIN <= 1e-37	FLT_MAX >= 1e37	FLT_DIG >= 6
double	12.3 0.12 12. .5 1E-3 1.8e5	DBL_MIN <= 1e-37	DBL_MAX >= 1e37	DBL_DIG >= 10
long double	12.3L 0.12l 12.l .5L 1E-3l 1.8e5L	<= mint double	>= mint double	>= mint double

Aritmetikai típusok konverziója

Egyoperandusú konverzió

- értékadáskor
- formális paraméter aktualizálásakor

Aritmetikai típusok konverziója

Kétoperandusú konverzió

- műveletvégzéskor

Típuskonverziók

Alapelv :

érték megőrzése, ha lehet

Típuskonverziók

Túlcsordulás esetén

a kapott érték elvileg definiálatlan

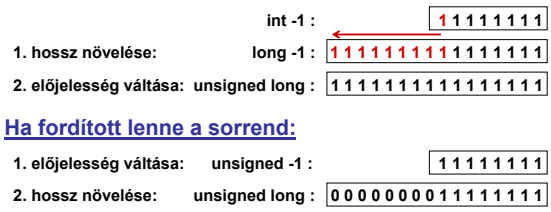
Típuskonverziók

Egészből egészbe

NINCS túlcsordulás jelzés

Típuskonverziók

Ha a hossz és az előjel is változik,
akkor ebben a sorrendben.



Ha fordított lenne a sorrend:

Mib l	Mibe	Eredmény
char, short	**int**	mindig, minden müvetelt előtt
rövidebb **int** pl. **short** **int**	hosszabb **int** ⇒ **int** ⇒ **long**	O.K.
rövidebb **unsigned** pl. **unsigned short** **unsigned**	hosszabb **unsigned** ⇒ **unsigned** ⇒ **unsigned long**	O.K.
hosszabb **int**	rövidebb **int**	túlcsordulhat (felső bitek elvesznek)
hosszabb **unsigned**	rövidebb **unsigned**	túlcsordulhat (felső bitek elvesznek)
rövidebb **+ int**	**unsigned**	O.K.
- int	**unsigned**	modulo 2^n, azaz **ebbe = ezt + 2^n** (**> ~_MAX** !!!)
unsigned	ugyanakkora **int**	túlcsordulhat
unsigned	hosszabb **int**	O.K.

Típuskonverziók

Lebegőpontosból lebegőpontosba

VAN túlcsordulás jelzés

Mib l	Mibe	Eredmény
rövidebb lebegőpontos pl.: **float ⇒** **double ⇒**	hosszabb lebegőpontos **double** **long double**	O.K.
hosszabb	rövidebb	túlcsordulhat, pontosság csökkenhet

Típuskonverziók

Egészből lebegőpontosba és vissza

VAN túlcsordulás jelzés

Mib l	Mibe	Eredmény
egész	lebegőpontos	ha lehet, a pontos érték megtartásával, ha nem: a legközelebbi két érték egyikére, ha nem lehet: túlcsordul
lebegőpontos	egész	törtrész elhagyásával, túlcsordulhat

Típuskonverziók

Lépések sorrendje kétoperandusú műveletek végrehajtása előtt

0. Ha egyik tömb vagy függvény, akkor mutatóvá konvertálódik
1. Egyoperandusú konverzió a hossz növelésére
2. A két operandus azonos típussá alakítása

Egyik operandus	Másik operandus	Közös, új típus
long double	bármi	**long double**
double	bármi	**double**
float	bármi	**float**
unsigned long	bármi	**unsigned long**
long	bármi (**int, unsigned**)	**long**
unsigned	bármi (**int**)	**unsigned**
int	bármi (**int**)	**int**

Típuskonverziók

Példa:

```
int a=3;
double b=2.3;
a=a*b;
```