

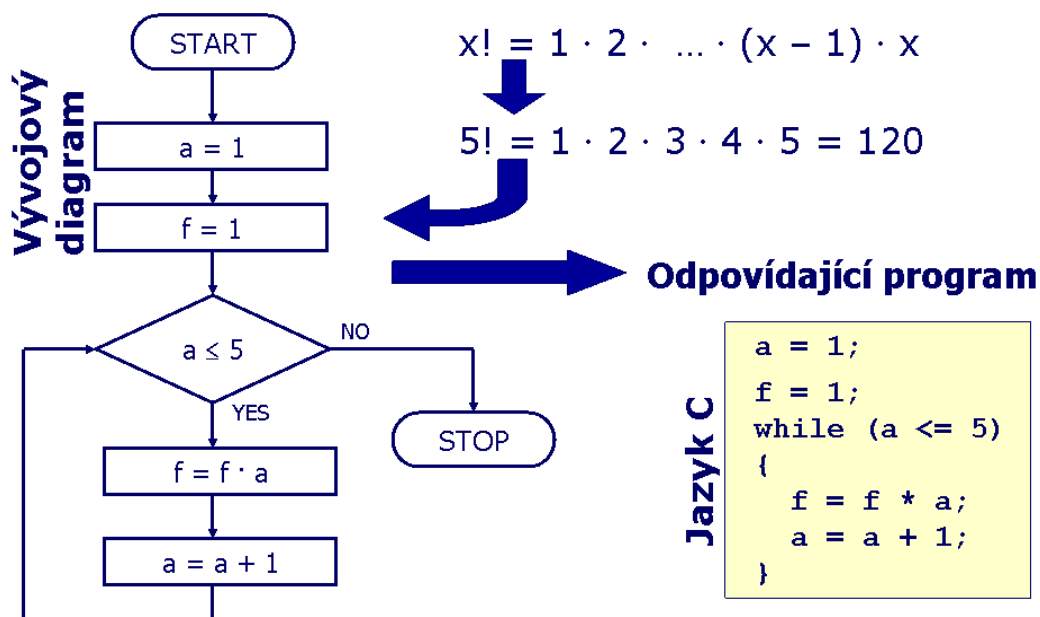
BU01 – Předpokládané teoretické znalosti

Základní pojmy

- **počítač** – „stroj na zpracování informací“
- **hardware** – technické vybavení počítače
- **software** – programové vybavení počítače
- **algoritmus** – postup pro řešení určité úlohy, který v každém kroku říká, jak dále postupovat
- **program** – algoritmus zapsaný v programovacím jazyce (posloupnost instrukcí)
- **programovací jazyk** – „umělý jazyk“ pro zápis programů (např. Pascal, C, C++, Java, Visual Basic, Fortran, ...)
- **instrukce** – předpis k provedení nějaké (většinou jednoduché) činnosti realizovaný přímo technickým vybavením počítače (např. přičtení jedničky, uložení hodnoty do paměti apod.)

Příklad algoritmu

Výpočet faktoriálu



Na konci běhu programu obsahuje proměnná f hodnotu 120.

Jednotky kapacity paměti

Bit

- nejmenší jednotka informace v paměti počítače
- označení **b**
- název vznikl z angl. binary digit (dvojkové číslo)
- buď 0 nebo 1

Bajt (Byte) – slabika

- skupina osmi bitů
- označení **B** (1 B = 8 b)

Word – slovo

- skupina několika bajtů (2 B, 4 B, 8 B apod.)

Předpony pro **násobky jednotek kapacity paměti** mají poněkud jiný význam než v soustavě SI:

$$\begin{aligned} 1 \text{ KB} &= 2^{10} \text{ B} = 1\,024 \text{ B}, \\ 1 \text{ MB} &= 2^{20} \text{ B} = 1\,048\,576 \text{ B} = 1\,024 \text{ KB}, \\ 1 \text{ GB} &= 2^{30} \text{ B} = 1\,073\,741\,824 \text{ B}, \\ 1 \text{ TB} &= 2^{40} \text{ B} = 1\,099\,511\,627\,780 \text{ B}. \end{aligned}$$

$$64 \text{ KB} = 65\,536 \text{ B}, 512 \text{ KB} = 0,5 \text{ MB}, \dots$$

Ne vždy však výrobci počítačových komponent (především pevných disků) tyto konvence dodržují. Mezinárodní elektrotechnická komise (IEC) doporučuje pro mocniny čísla 2 blízké hodnotám předpon SI používat nové předpony Ki, Mi, Gi, Ti atd. Tento mezinárodní standard má označení IEC 60027-2 a s platností od 1. 4. 2004 byl přejet do systému českých technických norem. V praxi se však tyto nově zavedené předpony moc nepoužívají.

Příklad: Použití jednotek kapacity paměti (z ceníku počítačových komponent):

- procesor Intel Xeon 5160 (Woodcrest), 3.00 GHz, Socket LGA771, FSB 1333 MHz, **4 MB** L2, Box
- DIMM Kingston DDR 2, **1024 MB**, PC2-3200 (400 MHz), ECC, Reg, Single Rank x 4, CL 3
- grafická karta Sapphire Radeon X1900 GT, **256 MB**, DDR 3, PCI-E 16x, VIVO, dual DVI-I
- HDD Seagate Barracuda 7200.10, **750 GB**, S-ATA II 3G, 7200 RPM, **16 MB**, RoHS
- DVD+R Verbatim, LightScribe, **4.7 GB**

Reprezentace dat v počítači

V počítači se obvykle ukládají odlišným způsobem data **numerická** (čísla) a data **nenumerická**. Čísla můžeme rozdělit na **celá** (integer) a **reálná** (real). Reálná čísla mohou mít buď **pevnou řádovou čárku** (fixed point) nebo **pohyblivou řádovou čárku** (floating point).

Příklad: Číslo **43** můžeme v počítači uložit například buď jako **celé číslo 43**, tj. 8bitovou binární kombinaci **00101011**, nebo jako **posloupnost znaků 4 a 3**, tj. binární posloupnost **00110100 00110011**, tedy dvojkové vyjádření ASCII hodnoty znaků 4 a 3 – viz dále.

Číselné soustavy

Obecně lze libovolné celé kladné číslo zapsat polynomem

$$a_n \cdot z^n + a_{n-1} \cdot z^{n-1} + \dots + a_0 \cdot z^0,$$

kde z je libovolné přirozené číslo větší než 1, tzv. **základ soustavy**, a koeficienty a_i jsou přirozená čísla, tzv. **číslíce soustavy**, splňující nerovnost $0 \leq a_i < z$.

Při zápisu čísla se obvykle používá zkrácený zápis $(a_n a_{n-1} \dots a_0)_z$, resp. $a_n a_{n-1} \dots a_0$, pokud nemůže dojít k pochybnosti, v jaké soustavě je číslo vyjádřeno.

Desítková (dekadická) číselná soustava

- jejím základem je číslo deset ($z = 10$)
- používá deset číslic (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
- v běžném životě nejpoužívanější
- čísla vyjadřujeme v jednotkách (10^0), desítkách (10^1), stovkách (10^2), tisících (10^3) atd.
- např. číslo **6307** můžeme vyjádřit jako:

$$\begin{array}{rcccccccc} 6 \text{ tisíc} & + & 3 \text{ sta} & + & 0 \text{ desítek} & + & 7 \text{ jednotek} & \\ 6 \cdot 1000 & + & 3 \cdot 100 & + & 0 \cdot 10 & + & 7 \cdot 1 & \\ 6 \cdot 10^3 & + & 3 \cdot 10^2 & + & 0 \cdot 10^1 & + & 7 \cdot 10^0 & = 6307, \end{array}$$

tedy $a_3 \cdot z^3 + a_2 \cdot z^2 + a_1 \cdot z^1 + a_0 \cdot z^0$, kde $z = 10$ (základ),
 $a_0 = 7, a_1 = 0, a_2 = 3, a_3 = 6, a_4, a_5, a_6, \dots = 0$ (číslíce)

Dvojková (binární) číselná soustava

- jejím základem je číslo dvě ($z = 2$)
- používá dvě číslice (0, 1)
- v oblasti výpočetní techniky nejpoužívanější
- desítkové číslo **11** můžeme vyjádřit jako dvojkové číslo **1011**:

$$\begin{array}{rcccccccc} 1 \cdot 2^3 & + & 0 \cdot 2^2 & + & 1 \cdot 2^1 & + & 1 \cdot 2^0 & \\ 1 \cdot 8 & + & 0 \cdot 4 & + & 1 \cdot 2 & + & 1 \cdot 1 & = 11 \dots \text{dekadicky} \end{array}$$

- pokud by mohlo dojít k nejasnostem, v jaké soustavě je dané číslo zapsáno, používá se forma zápisu $(1011)_2 = (11)_{10}$, což čteme jako „*dvojkové (binární) číslo jedna nula jedna jedna je rovno desítkovému (dekadickému) číslu jedenáct*“.

Šestnáctková (hexadecimální) soustava

- jejím základem je číslo šestnáct ($z = 16$)
- používá šestnáct číslic (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F), přičemž písmena A, B, C, D, E, F odpovídají po řadě číslům 10, 11, 12, 13, 14, 15
- v oblasti výpočetní techniky slouží především ke zjednodušení zápisu dvojkových čísel
- desítkové číslo **967** můžeme vyjádřit jako šestnáctkové číslo **3C7**:

$$\begin{array}{rcccccccc} 3 \cdot 16^2 & + & C \cdot 16^1 & + & 7 \cdot 16^0 & \\ 3 \cdot 256 & + & 12 \cdot 16 & + & 7 \cdot 1 & = 967 \dots \text{dekadicky} \end{array}$$

- zápis $(3C7)_{16} = (967)_{10}$ čteme „*šestnáctkové (hexadecimální) číslo tři cé sedm je rovno desítkovému (dekadickému) číslu devět set šedesát sedm*“.

Převody mezi číselnými soustavami

Příklad: Převed'te čísla **10** a **23** z desítkové soustavy do dvojkové.

Požadované desítkové číslo postupně dělíme dvěma, zapíšeme zbytek a každý výsledek opět dělíme dvěma, až dostaneme nulový podíl. První číslicí ve dvojkové soustavě bude zbytek získaný posledním dělením.

výsledek po dělení 2	zbytek
$10 : 2 = 5$	0
$5 : 2 = 2$	1
$2 : 2 = 1$	0
$1 : 2 = 0$	1

$$(10)_{10} = (1010)_2$$

výsledek po dělení 2	zbytek
$23 : 2 = 11$	1
$11 : 2 = 5$	1
$5 : 2 = 2$	1
$2 : 2 = 1$	0
$1 : 2 = 0$	1

$$(23)_{10} = (10111)_2$$

Příklad: Převed'te dvojkové číslo **101110** do desítkové soustavy.

Dvojkové číslo **101110** můžeme zapsat jako

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 \\ 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 46$$

$$(101110)_2 = (46)_{10}$$

Příklad: Převed'te desítkové číslo **586** do šestnáctkové soustavy.

Desítkové číslo 586 si nejdříve převedeme na dvojkové. Poté si bity dvojkového čísla zprava rozdělíme na čtveřice, které budou představovat jednotlivé číslice šestnáctkového čísla (případně chybějící bity u nejlevější čtveřice doplníme nulami). Takto získané čtveřice postupně převedeme na desítkové ekvivalenty, čísla větší než 9 nahradíme odpovídajícími písmeny šestnáctkové soustavy:

$$10 = \mathbf{A}, \quad 11 = \mathbf{B}, \quad 12 = \mathbf{C}, \\ 13 = \mathbf{D}, \quad 14 = \mathbf{E}, \quad 15 = \mathbf{F}.$$

Tak získáme zápis čísla v hexadecimální soustavě.

výsledek po dělení 2	zbytek
$586 : 2 = 293$	0
$293 : 2 = 146$	1
$146 : 2 = 73$	0
$73 : 2 = 36$	1
$36 : 2 = 18$	0
$18 : 2 = 9$	0
$9 : 2 = 4$	1
$4 : 2 = 2$	0
$2 : 2 = 1$	0
$1 : 2 = 0$	1

$$(586)_{10} = (1001001010)_2$$

$$\begin{array}{ccc} 10 & 0100 & 1010 \\ 0010 & 0100 & 1010 \\ 2 & 4 & 10 \\ 2 & 4 & \mathbf{A} \end{array}$$

$$(586)_{10} = (24\mathbf{A})_{16}$$

Příklad: Převed'te hexadecimální číslo **2AC7** na dekadické.

Číslo $(2AC7)_{16}$ zapíšeme ve tvaru:

$$\begin{array}{rcccccccc}
 2 \cdot 16^3 & + & A \cdot 16^2 & + & C \cdot 16^1 & + & 7 \cdot 16^0 & \\
 2 \cdot 16^3 & + & 10 \cdot 16^2 & + & 12 \cdot 16^1 & + & 7 \cdot 16^0 & \\
 2 \cdot 4096 & + & 10 \cdot 256 & + & 12 \cdot 16 & + & 7 \cdot 1 & = 10951
 \end{array}$$

$$(2AC7)_{16} = (10951)_{10}$$

■

Vyjádření čísel v desítkové, dvojkové a šestnáctkové soustavě

Dec.	Bin.	Hex.	Dec.	Bin.	Hex.	Dec.	Bin.	Hex.	Dec.	Bin.	Hex.
0	00000000	0	16	00010000	10	32	00100000	20	48	00110000	30
1	00000001	1	17	00010001	11	33	00100001	21	49	00110001	31
2	00000010	2	18	00010010	12	34	00100010	22	50	00110010	32
3	00000011	3	19	00010011	13	35	00100011	23	51	00110011	33
4	00000100	4	20	00010100	14	36	00100100	24	52	00110100	34
5	00000101	5	21	00010101	15	37	00100101	25	53	00110101	35
6	00000110	6	22	00010110	16	38	00100110	26	54	00110110	36
7	00000111	7	23	00010111	17	39	00100111	27	55	00110111	37
8	00001000	8	24	00011000	18	40	00101000	28	56	00111000	38
9	00001001	9	25	00011001	19	41	00101001	29	57	00111001	39
10	00001010	A	26	00011010	1A	42	00101010	2A	58	00111010	3A
11	00001011	B	27	00011011	1B	43	00101011	2B	59	00111011	3B
12	00001100	C	28	00011100	1C	44	00101100	2C	60	00111100	3C
13	00001101	D	29	00011101	1D	45	00101101	2D	61	00111101	3D
14	00001110	E	30	00011110	1E	46	00101110	2E	62	00111110	3E
15	00001111	F	31	00011111	1F	47	00101111	2F	63	00111111	3F
								

Pro převod čísel mezi jednotlivými číselnými soustavami můžeme využít i kalkulačku, kterou nalezneme ve skupině *Příslušenství* operačního systému *MS Windows*. Nebudeme-li mít v levé části kalkulačky k dispozici sadu přepínačích tlačítek označených *Hex*, *Dec*, *Oct* a *Bin*, vybereme si z nabídky posloupnost příkazů *Zobrazit – Vědecká*.

ASCII tabulka

- zkratka z angl. **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
- standardní sada znaků definovaná v roce 1968
- původně navržena jako 7-bitová ($2^7 = 128$ znaků), nyní se používá jako 8bitová ($2^8 = 256$ znaků)
- dolních 128 znaků je jednotných po celém světě – **základní část**
- horních 128 znaků se může lišit podle národního prostředí (znaky s diakritikou apod.) – **rozšířená část**, z čehož vyplývají velké problémy se zobrazováním znaků různých národních abeced (včetně češtiny)

Příklad: Chceme-li z české klávesnice napsat znak @, který se na ní nenachází, držíme levý Alt a na numerické části klávesnice napíšeme dekadický ASCII kód tohoto znaku, tj. **64**.

■

Základní část ASCII tabulky (znaky 0 – 127)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.asciitable.com

ASCII tabulka – problémy s češtinou

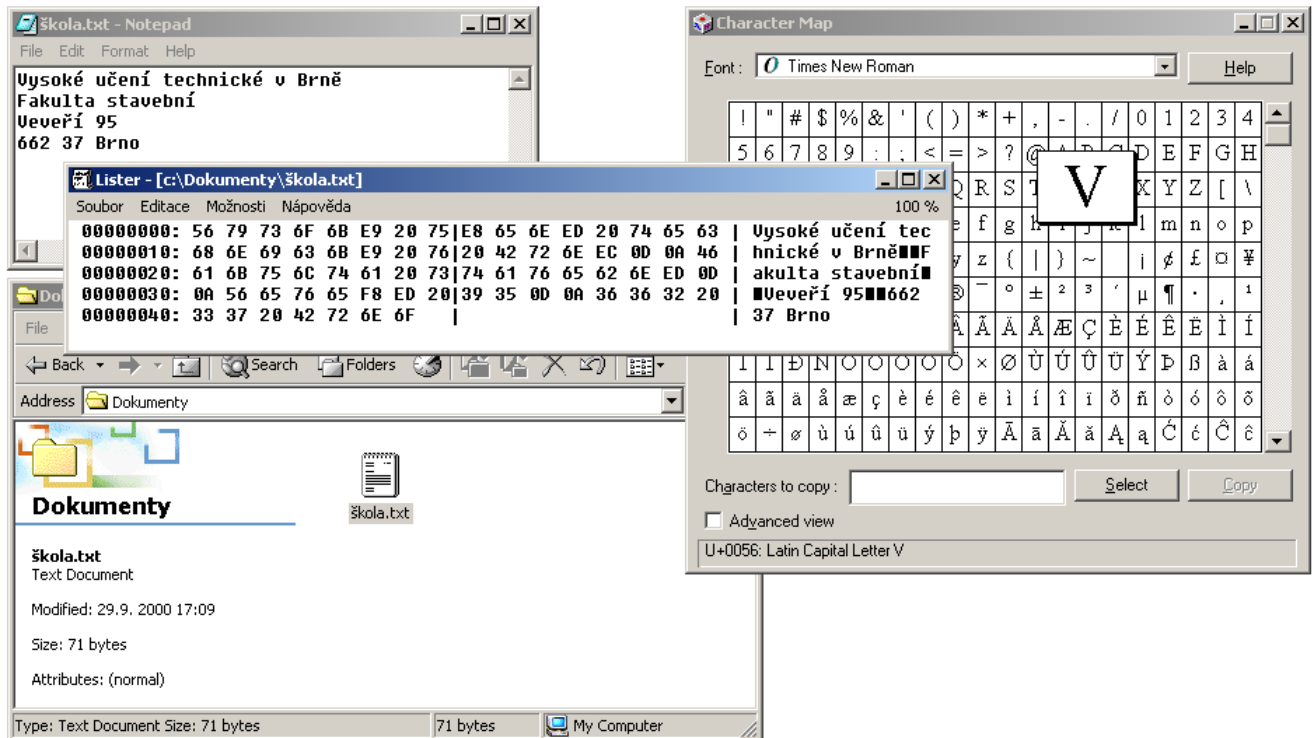
- pro češtinu existuje několik způsobů kódování (znakových sad):
 - ISO-8859-2 (ISO Latin 2),
 - Windows 1250 (CP1250),
 - CP852 (PC Latin 2),
 - bratří Kamenických,
 - KOI8-CS,
 - ...
- všechny tyto znakové sady se liší horní polovinou ASCII tabulky (znaky 128 – 255) a nejsou tedy navzájem kompatibilní,
- další zajímavé informace naleznete na www.cestina.cz.

Správné nastavení kódové stránky si můžeme vyzkoušet např. na typicky české větě **Příliš žluťoučký kůň úpěl ďábelské ódy**. Použijeme-li pro interpretaci binárně uložených znaků nesprávnou kódovou tabulku, nebude text plně čitelný.

Příklad: Věta **Příliš žluťoučký kůň úpěl ďábelské ódy** uložená v kódové stránce **CP1250**, ale zobrazená v kódové stránce **ISO Latin 2: Příliš žluťoučký kůň úpěl ďábelské ódy**. Použijeme-li pro zobrazení českého textu naprosto nevhodnou znakovou sadu (např. **CP1251** použitelnou pro azbuku), obdržíme text: **Ршнлль ђлуќоуќкэ куџ ѓрмл ђђбелскй уды**. Všimneme si, že korektně jsou zobrazeny pouze znaky bez diakritických znamének, tj. znaky ze spodní části ASCII tabulky, která je společná pro všechny znakové sady.

Příklad uložení dat v neformátovaném textovém souboru

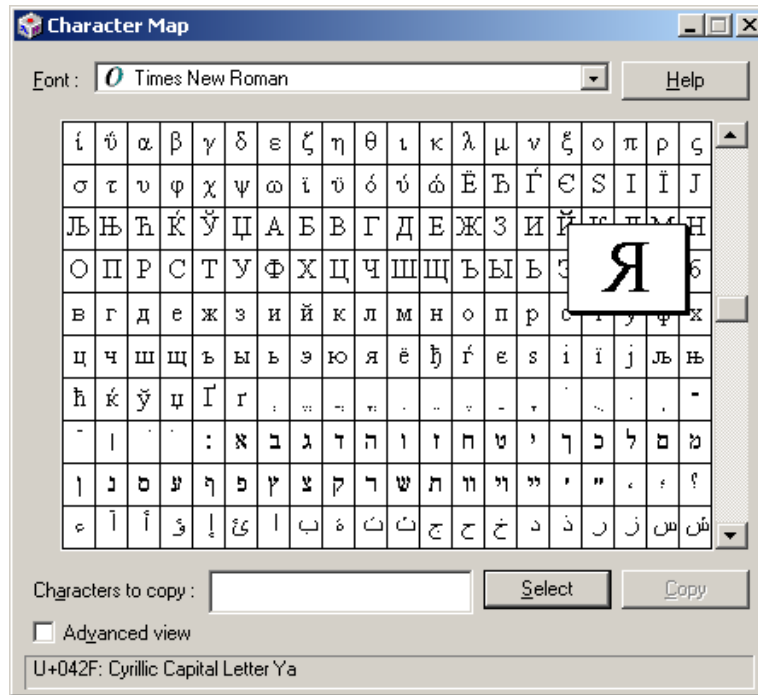
- použitá kódová stránka Windows 1250 (CP 1250),
- v prostém textu jeden znak zabírá jeden bajt,
- konec řádku jsou zde dva znaky – CR, LF,
- ASCII hodnoty znaků jsou na obrázku uvedeny v hexadecimální soustavě.



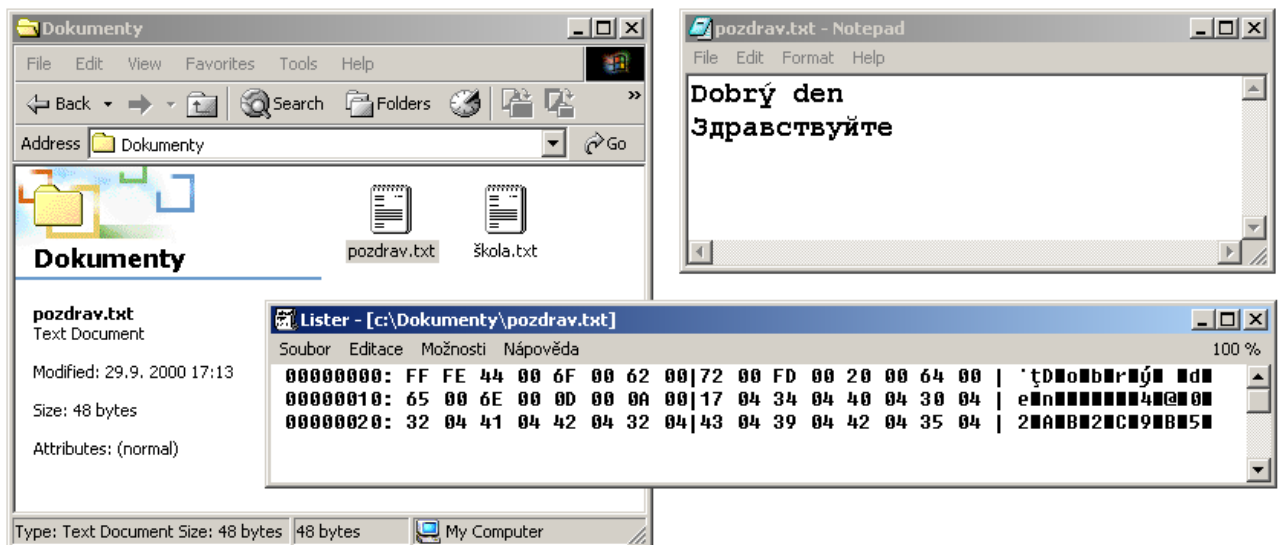
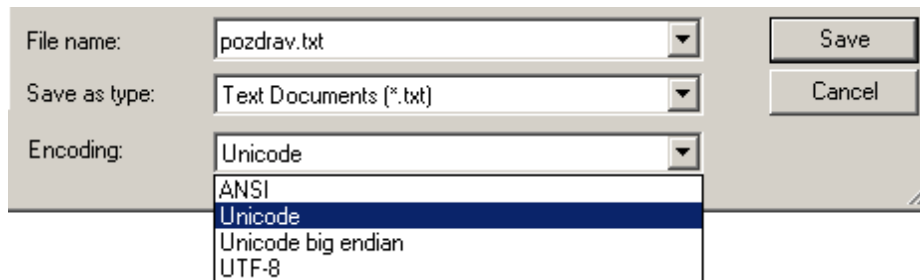
Protože jde o klasický neformátovaný textový soubor, snadno nahlédneme, že například první písmeno textu **V** je uloženo jako **hexadecimální číslo 56**, což je v souladu s ASCII hodnotou tohoto znaku. Obdobným způsobem číslo orientační **95** je uloženo jako **posloupnost znaků 9 a 5**, tj. **hexadecimální čísla 39 a 35** odpovídající ASCII hodnotám těchto znaků. Každou **mezeru** identifikujeme jako **hexadecimální číslo 20** (space), každý **konec řádku** (Enter) jako dvojici **0D 0A** (CR LF).

Unicode

- moderní standard kódování znaků, který používá 16 bitů na jeden znak ($2^{16} = 65\,536$ různých znaků), čímž se pokryjí znaky většiny jazyků na světě (ruština, arabština, ...),
- řeší problém globální výměny dat,
- nevýhody: dvojnásobná délka textu (1 B → 2 B) a tím i trochu pomalejší zpracování dat, větší znaková sada (až 256krát), problémy se zpětnou (8bitovou) kompatibilitou.



- aplikace *Notepad* (*Poznámkový blok*) standardně ukládá soubory ve formátu ANSI; při ukládání souboru si proto musíme vybrat Unicode



Kontrolní otázky

Jaký je rozdíl mezi bitem a bajtem?

Převeďte na jednotky uvedené v závorkách: 1 MB (KB), 2 KB (B), 64 b (B), 100 B (b), 512 B (KB), 0,5 GB (MB).

Převeďte desítkové číslo ... do dvojkové soustavy.

Převeďte dvojkové číslo ... do desítkové soustavy.

Převeďte desítkové číslo ... do šestnáctkové soustavy.

Převeďte šestnáctkové číslo ... do desítkové soustavy.

Převeďte dvojkové číslo ... do šestnáctkové soustavy.

Převeďte šestnáctkové číslo ... do dvojkové soustavy.

Jak ve dvojkové soustavě snadno rozpoznáme sudá a lichá čísla?

Jaké maximální číslo můžeme zobrazit na n bitech?

Jaký tvar mají ve dvojkové soustavě čísla, která jsou mocninou dvou?

K čemu slouží ASCII tabulka?

V čem spočívají problémy zobrazování českých znaků na počítači?

Vyjmenujte alespoň tři znakové sady použitelné pro kódování češtiny.

Co je to Unicode?

Uveďte výhody a nevýhody používání Unicode.

Užitím ASCII tabulky запиšte slovo „Ahoj“ v binární formě.

K čemu slouží znaky CR, LF?

Jak v prostředí webového prohlížeče změním kódovou stránku zobrazeného textu?

BU01 – Předpokládané praktické znalosti

Operační systém MS Windows

- manipulace s okny, přepínání běžících úloh, přepínání mezi nainstalovanými klávesnicemi, ...
- vytváření, mazání, kopírování a přesunování složek a souborů,
- programy skupiny Accessories (Příslušenství),
- clipboard (schránka) a jeho funkce.

Práce v prostředí počítačové sítě

- standardní prohlížeč webových stránek a jeho nastavení,
- vyhledávací servery a jejich využití,
- práce s elektronickou poštou.