

Kódování ornamentů

V minulé části seriálu jsme popsali způsob kódování standardních písem. V tomto odstavci popíšeme způsob kódování písem, která neobsahují latinkovou abecedu, ale různé piktogramy, ikony, ornamenty a podobně. Jedná se například o písma Symbol, Zapf Dingbats a podobně.

Při práci s ornamenty vzniká jistá potíž, daná schopnostmi (či spíše neschopnostmi) běžných jednobajtově orientovaných systémů a aplikací: operační systémy či aplikace totiž nepředpokládají práci s více než jednou znakovou sadou; ornamenty přitom ale nejsou nic jiného než právě jiná znaková sada. Jinými slovy, pokud v takovémto systému napíšete několik znaků písmem Symbol, tak vám systém nijak nezabrání napsaný text označit a například převést na verzálky (velká písmena), což je ale v případě symbolů z písma Symbol nesmysl: o znacích jako jsou •, ≠, £, ÷ nebo « nemůžeme říci, jedná-li se o verzálky nebo minusky (malá písmena). Podobně, následující čtyři ornamenty ze Zapf Dingbats ☼*□* určitě netvoří slovo; přesto ale, poklepete-li na ně, označí se jako jedno slovo (což je způsobeno tím, že jejich kódy odpovídají písmenám „ahoj“). Lze tedy očekávat, že při tisku ovladač tiskárny tato písma správně nezakóduje, protože nepředpokládá práci s více než jednou znakovou sadou, jinými slovy nezná sadu liter (tedy jejich jména), která je v daném písmu obsažena. Je snad zřejmé, že se litery nejmenují stejnými jmény, jako litery ze standardních písem.

V případě ornamentů proto firma Adobe použila jiný způsob kódování; pro každé písmo s ornamenty je definováno jeho vlastní kódování a příslušný kódovací vektor je umístěn přímo v písmu. Tiskový ovladač je potom vhodným způsobem informován, že dotyčné písmo nemá při tisku překódovávat do vlastní znakové sady, ale ponechat v jeho vlastním kódování. Většinou se jedná o jeden bit (na počítačích Macintosh například v kufu s písmy); jindy může být použit poněkud inteligentnější způsob (častokrát pod aplikacemi v systému Unix), kdy se o překódování rozhodne až v tiskárně na základě toho, je-li písmo ve StandardEncoding či nikoli (toto rozhodnutí je přitom obsaženo v postscriptovém kódu, který ovladač do tiskárny zavede). Poznamenejme ale, že Macintosh může být v tomto případě výjimkou, neboť umožňuje písma s ornamenty překódovat; písmo Symbol na počítači Macintosh např. obsahuje znak 🍏 na pozici 240, která je ve Windows neobsazena.

Výše uvedené řešení má ovšem ten důsledek, že písmo s ornamenty je ve stejném kódování na všech platformách. Jinými slovy, znak ^ z písma Symbol má vždy kód 94 (tedy stejný, jako litera „^“ v ASCII kódování), znak ™ má vždy kód 228, což odpovídá písmenu „š“ v kódování Macintosh CE, znaku „%“ v kódování Macintosh Roman a písmenu „ä“ v kódování Windows, Windows EE i ISO Latin 1 a Latin 2. Jinými slovy, tento znak se vždy píše na klávesnici jinak.

Shodnost kódování symbolů na různých platformách má ještě ten důsledek, že v těchto písmech nejsou většinou obsazeny pozice 128 až 159, které na některých platformách nejsou dostupné.

Kódování středoevropských a nelatinkových písem

Zmiňme se nejprve o kódování středoevropských (CE) písem a cyrilice. Na většině počítačů kromě počítače Macintosh je totiž situace obdobná, jako v případě písem s ornamenty: operační systém nebo aplikace pracuje s jednou znakovou sadou a text psaný např. CE písmem (tedy česky) chápou úplně stejně, jako text psaný v původní znakové sadě. Důsledkem toho je, že se například při poklepání na slovo neoznačí toto slovo celé nebo se označí více než toto slovo, nebo například to, že nefunguje správně

převod mezi verzálkami a minuskami, a tak dále a tak dále. Zkuste například v české verzi napsat písmem Arial text a³x (exponent je znak s kódem 179) a poté zklepat na písmeno a. Označí se (chybně) celý výraz. Nebo obráceně, zkuste v americké verzi Windows napsat slovo „žížala“ v písmu Arial CE a potom a něj zklepat. Označí se jen část „ala“. Jedinou výjimkou v tomto případě je Macintosh a jeho systém znakových sad. Firma Adobe proto musela zvolit stejný postup, který použila u písem s ornamenty, totiž ten, že písmo je napevno zakódováno a tiskový ovladač ho při tisku nepřekódovává, protože to zkrátka neumí. Písmo je přitom zakódováno do odpovídající znakové sady Windows (CP 1250 = Windows EE, resp. CP 1251 pro cyrilici), a to jak ve verzi pro Windows, tak ve verzi pro Macintosh(!). Na počítači Macintosh je totiž tiskový ovladač schopen písmo překódovat do své vlastní znakové sady Macintosh CE (nebo Macintosh Cyrillic), a to i v případě, že písmo není ve StandardEncoding. Jinými slovy, při tisku z počítače Macintosh ovladač písmo v kódování Windows EE překóduje do sady Macintosh CE a při tisku z Windows zůstane písmo v kódování Windows EE (které je v něm obsaženo), neboť ovladač Windows ho překódovat prostě neumí. Tím je tedy dosaženo toho, že písmo je na obou platformách shodné a lze je po zavedení na disk tiskárny sdílet oběma platformami a přitom z každé v jiném kódování.

Pro systémy pracující pod systémem Unix je situace poněkud složitější, například proto, že pro kódování české abecedy se používá několik různých kódů, častokrát i v rámci jednoho systému (jiný kód pro různé aplikace v textovém režimu a jiný pod X Window). V současné době proto zatím neexistuje univerzální řešení a případy jednotlivých systémů tady nebudeme podrobněji rozebírat.

Kódování ostatních nelatinkových písem, jako je například japonština nebo čínština, je opět založeno na jménech liter a kódovacím vektoru a poskytuje podobné vlastnosti, jako v případě písem standardních. Popis znakových sad, používaných pro japonštinu, stejně tak jako popis poněkud složitější struktury písem, by byl značně rozsáhlý a překročil by zaměření tohoto článku. Zájemce nalezne podrobnější informace například v PostScript SDK firmy Adobe.