

Epoc (dokončení z minulého čísla)

Vývoj pro Epoc

V minulém Chipu jste se mohli seznámit se základními rysy operačního systému Epoc, používaného pro mobilní zařízení. Dnes článek dokončíme popisem možností vývoje aplikací pro tento systém.

Epoc své programátory nijak nehýčká; pro uživatele špičkového API, jakým je OpenStep nebo Apple Cocoa, je jeho API velmi strohé. Většina programátorů však má zkušenosti jen s nesrovnatelně horšími posixovými nebo windowsovými prostředím – z jejich pohledu bude programování pro Epoc poměrně pohodlné.

Součástí vývojového prostředí Epocu je poměrně široká podpora standardů. Základní programovací jazyky jsou C/C++ a Java, k dispozici je i (ne zcela kompletní) API Posix. Co je to však všechno platné, když "stolní" aplikace v naprosté většině případů z příčin rozebranych v první části článku na mobilní systémy portovat nelze.

Patrně ideální by bylo standardní API rozšířené o specifické služby, odpovídající mobilnímu prostředí. Epoc je víceméně objektový operační systém; jediné dnes standardizované objektové API je OpenStep, ten však bohužel v Psionu asi neznají. Proto navrhli proprietární API, založené na C++, tedy víceméně neobjektové, protože podpora objektů v C++ je dost tristní.

I přes tuto nevýhodu platí pro API Epocu zhruba totéž co pro celý operační systém: není ani zdaleka dokonalé, je ale lepší než API systémů jako PalmOS nebo Windows CE. I přes omezení, daná jazykem C++, dodržuje API Epocu alespoň zásadní objektovou strukturu a i jeho služby jsou na dost vysoké úrovni.

Firma X.soft sice pracuje na otevřeném SDK pro Epoc, založeném na GNUStepu, to je však dlouhodobý projekt, takže v současnosti i v střednědobých výhledech pořád zůstává s přehledem nejlepším "mobilním" API standardní API Epocu.

Programovací jazyky

Základním programovacím jazykem Epocu je C++, v němž je sám Epoc napsán. Podobně jako například u BeOS to přináší řadu nepříjemných omezení, pro "mobilní" Epoc to však nese i některé výhody.

C++ je patrně nejhorší z dnes běžně užívaných jazyků – uvedme jen orientačně několik jeho hlavních nevýhod:

- * s jeho nepřehledností snad může soutěžit jen PL-1 neblahé paměti; proto vyžaduje relativně dlouhou dobu na naučení;

- * nepodporuje korektně práci s objekty (např. třída je pro něj jen typ, nelze ji uložit do proměnné nebo předat jako argument; metody jsou volány staticky; vůbec nemá protokoly...);

- * nemá prakticky žádnou běhovou ochranu (např. volání metody nesprávného nebo neexistujícího objektu vede k pádu programu bez informace o příčině chyby);

- * vícenásobná dědičnost, overloading operátorů a templates přesnadno vedou k dokonale nečitelným programům – tedy obtížně laditelným a s extrémně drahou údržbou.

Na druhou stranu má C++ i některé výhody. Především, z nepochopitelných příčin (kvůli jeho kvalitám to rozhodně není), je velmi rozšířen: ačkoli dobrých programátorů v C++ je velmi málo, jeho základy zná téměř každý a skoro každý v něm dokáže "něco nasmolit". Právě díky jeho největší nevýhodě – statickému přístupu k objektům – je přeložený kód relativně velmi efektivní (statické volání metody v C++ je v průměru 1,5krát až 2krát rychlejší než předání dynamické zprávy v Objective C): u mobilních systémů s jejich nízkým výpočetním výkonem to má smysl. Tvůrci Epocu se alespoň těm nejhorším vlastnostem C++ dokázali vyhnout: overloading operátorů se prakticky nevyužívá, vícenásobná dědičnost slouží jen pro simulaci protokolů, templates se využívají pouze pro vylepšení typové kontroly.

V určitém smyslu srovnatelná s unixovými operačními systémy je Java. Je to velmi slušný objektový programovací jazyk, odstraňující většinu chyb jazyka C++ a trpící trochu jen nešťastnou "tečkovou" syntaxí; to ale vyjma snížení čitelnosti zdrojového kódu není zásadní nevýhoda. Zásadní nevýhodou pro mobilní systémy je však její náročnost na výpočetní výkon. Jakkoli interpreter Javy v Epocu patří mezi nejlepší, pro implementaci náročnějších algoritmů je Java nepoužitelná. Další omezení jsou dána tím, že sám Epoc je bohužel implementován v C++ a že z toho, čemu se tam říká třída či objekt, udělat plnohodnotnou javovskou třídu nebo objekt prostě nejde. Proto je z Javy trochu komplikovaný přístup k systémovým službám. Přesto lze předpokládat, že Java určitě znamená budoucnost programování v Epocu – vyjma jednoduchých a nenáročných aplikací, na které stačí dnešní výkon, však budoucnost poměrně vzdálenou. Prozatím je pro plnohodnotné aplikace bohužel nutné zůstat u C++.

Třetím a posledním jazykem, který Epoc API standardně podporuje, je OPL, což je proprietární varianta strukturovaného Basicu. OPL je interpretovaný jazyk a jako jediný umožňuje vývoj přímo v Epocu (pro vývoj v Javě nebo v C++ musíme využívat křížové SDK na stolním počítači). OPL má několik výhod, které jej předurčují pro rychlou tvorbu jednoduchých aplikací: sám jazyk je prostinký a dá se jej naučit za pár dní (zkušenější programátor se jej naučí za pár hodin). Jeho interpreter je velmi rychlý – aplikace psané v OPL

jsou o poznání rychlejší než aplikace psané v Javě, i když nemohou soutěžit s aplikacemi psanými v kompilovaném C++.

Nakonec bych se rád velmi stručně zmínil o Objective C: stručně proto, že v současnosti podporované není. Je to však nesmírná škoda, protože jde o ideální kompromis: jazyk s možnostmi a elegancí Javy (v řadě detailů dokonce lepší a s efektivitou blízkou se C++. Otevřené SDK, na němž pracuje X.soft, bude Objective C samozřejmě podporovat. Bude také podporovat tvorbu projektů využívajících pro různé části různých jazyků; i to je nyní dost problematické.

C++ SDK

Vzhledem k uvedenému se dále zaměřím jen na SDK C++, jehož základní koncepci navrhli v Psionu bohužel dost nešťastně. Předpokládá se totiž, že aplikace bude nejprve odladěna v tzv. "emulátoru" Epocu. Tato myšlenka má dvě zásadní nevýhody: předně, emulátor je k dispozici jen pro Windows, takže uživatelé jiného prostředí mají smůlu. Co hůř, vůbec nejde o emulátor; je to jen port Epocu, hostující uvnitř Windows a využívající jejich zdroje. Vzhledem k tomu běží celý emulátor nad "cizí" architekturou; proto odlišné jsou také formáty jeho spustitelných souborů, jiný je strojový kód uvnitř nich, a dokonce se liší i překladač, který se pro jejich tvorbu používá! Zatímco pro tvorbu cílových aplikací slouží vcelku rozumné GNU C++, aplikace pro "emulátor" se vytvářejí v MS Visual C++.

Epoc SDK proto dokáže vytvářet aplikace tří typů: tzv. WINS, WINC a MARM. WINS jsou právě aplikace pro zmíněný emulátor; WINC jsou "epocovské" aplikace, které lze spouštět přímo pod Windows. Na rozdíl od emulátoru však využívají GUI Windows – ne Epocu – a mají neomezený přístup k windowsovským zdrojům. Hlavním účelem této služby je podpora konverzí, na kterou se podrobně podíváme níže. Konečně MARM jsou aplikace určené pro práci na cílové architektuře (v současnosti počítače s Epocem využívají procesory ARM, odtud MARM).

Paradoxní je, že tvůrci této "potvornosti" uvádějí, že chtěli "umožnit využití skvělého debuggeru" z MSVC. Přitom kterýkoli uživatel gdb, který musel pracovat v MSVC, jistě zažil šok: sice je vše v barevných okénkách, ale vůbec nic to neumí – dokonce nelze ani do okénka pro zobrazení hodnoty zapsat libovolný výraz!

V praxi je proto nejrozmumnější vytvářet aplikace přímo pro cílovou platformu. K tomu slouží GNU C++; standardně bohužel opět v jakési windowsovské mutaci, jež občas přináší podivné problémy (např. nelze překládat zdrojové kódy rozložené na více disků!). I přesto je to téměř optimální – stačí využít standardní distribuci GNU C++, a skoro bychom mohli pro Epoc vyvíjet kdekoliv. Skoro proto, že SDK Epocu využívá několik proprietárních konverzních programů, jež je zapotřebí nejprve upravit pro platformu, na níž chceme mobilní aplikace vyvíjet.

Tato "neportabilita" vývojového prostředí je jen dočasná. Horší problém je bohužel s laděním: zatímco samotná tvorba epocovských aplikací v rámci libovolné architektury je otázkou poměrně krátkého času, je v současnosti naprosto nejasné, jak takové aplikace ladit. Epoc nabízí pro ladění "emulátor", ten je ale použitelný jen pod Windows a není přenositelný (byl by, pokud bychom měli jeho zdrojové texty; ty ovšem k dispozici nejsou a těžko kdy budou).

Bude tedy zapotřebí nejspíše připravit doplněk Epocu, umožňující remote ladění s využitím gdb a sériového kabelu, nebo napsat skutečný emulátor. Ani jedno bohužel není jednoduché; tvorba emulátoru navíc naráží na nedostatečně dokumentovaný hardware nejrozumnějších přístrojů pracujících pod Epocem. Zdá se tedy, že po nějakou dobu bude nutné se při ladění spokojit s klasickými asserty a kontrolními výpisy.

Kromě GNU C++, dokumentace a řady příkladů už toho standardní vývojové prostředí moc nenabízí: nalezneme v něm jen několik perlových scriptů a windowsových "batchů", které zajišťují tvorbu a zpracování projektů. Tento systém je dost nepohodlný a má řadu umělých omezení (např. složku, ve které projekt leží, nelze přejmenovat, aniž bychom zároveň změnili projekt); proto ani nemá smysl uvažovat o jeho portu. Namísto toho X.soft připravuje SDK, využívající standardních makefiles a GUI prostředků včetně vizuálního programování (o kterém se Epocu v současnosti bohužel ani nezdá). Předběžné informace o novém SDK lze nalézt na www.ocs.cz/text/XSdk.

Přehled služeb Epocu

Všechny služby jsou k dispozici prostřednictvím tříd C++ z dynamicky zaváděných knihoven – toto řešení má výhody i nevýhody. Nevýhody jsou opět dány omezeními jazyka C++. Jelikož jde o statický jazyk, nejsou zde k dispozici ani kategorie, o možnosti dynamicky nahradit jednu třídu jinou už vůbec nemluví. Proto je problém s doplňky a úpravami systému.

Na druhou stranu je to nejkorektnější možné řešení a patrně nepohodlnější i z hlediska programátora. Vzhledem k tomu, že standardní knihovny jsou v paměti ROM (díky kvalitní správě paměti Epocu není třeba je zavádět, ale jsou plně a kdykoli k dispozici přímo na svých "romkových" adresách), je spouštění aplikací velmi rychlé. Proto si Epoc může snadno dovolit objektovou aplikační architekturu (viz dále).

Base

Skupina tříd, sdružená pod pojmem Base, nabízí základní služby pro "engine" aplikací: nalezneme zde kontejnerové objekty, služby pro práci s buffery a textovými řetězci (API podporuje i UNICODE, avšak

implementace dosud není tak daleko), poměrně luxusní a efektivní systém odchycení a zpracování chybových stavů a další služby.

File server

Základní služby pro práci se soubory až na mírně odlišné API velmi přesně odpovídají klasickým posixovým souborům. Epoc však nabízí mnohem luxusnější služby než naprostá většina ostatních prostředí pro práci s obecnými streamy. Architektura tzv. "stream stores" totiž podporuje rekurzivní vnořování streamů nebo ukládání více streamů do společného kontejneru. To je samozřejmě ideální pro vkládání dokumentů (typu "obrázek vložený v tabulce vložené v textovém dokumentu") a nabízí to velmi luxusní možnosti i pro zapouzdření dat. Představte si textový editor; je velmi přirozené – a z hlediska programátora i velmi pohodlné – uložit čistý text do jednoho streamu, formátovací informace do druhého, do třetího například jméno autora a do čtvrtého nastavení uživatelského rozhraní (např. zoom)... Streamy se mohou skládat i "do série" – je například velmi snadné pracovat se streamem kódovaným pomocí hesla: prostě "nad" streamem se skutečnými daty otevřeme stream, který se stará o šifrování, a k datům přistupujeme jeho prostřednictvím.

Databáze

Epoc standardně obsahuje velmi luxusní databázové API a trochu méně luxusní implementaci databázových služeb. API nabízí skvělé služby pro relační databázi včetně přístupu s využitím standardních SQL příkazů. Konkrétní implementace nabízí trochu omezený subset SQL a je realizována jako sada služeb na straně klienta pro přístup k databázovým souborům. Zatím tedy není k dispozici plnohodnotný server se sdíleným přístupem k datům. I tak jsou ale databázové služby Epocu prakticky srovnatelné například s OpenBase Lite a mnohonásobně převyšují služby databází z wintelu (např. FoxPro).

Aplikační architektura

Architektura aplikací Epocu je poměrně důslednou implementací plně objektového pohledu, kdy vlastně aplikace jako takové neexistují, ale systém obsahuje služby pro zpracování různých druhů dokumentů. To samozřejmě přináší určité problémy (např. s otevíráním jednoho dokumentu v různých aplikacích), ale významně to usnadňuje život laickému uživateli, který se o výpočetní techniku nezajímá a studovat ji nechce – chce prostě jen používat ten či onen přístroj. To dnes již platí dokonce i o většině uživatelů počítačů; docela jistě to platí o drtivé většině uživatelů mobilních přístrojů, od telefonů až po organizéry.

Uživatel tedy většinou nepracuje s "aplikacemi", ale namísto toho jen zavírá a otevírá "dokumenty". Pojem "aplikace" – a některé na něj vázané prvky uživatelského rozhraní – zůstaly zachovány především kvůli aplikacím, jež s dokumenty nepracují (např. hry). Dokumentem aplikace je vždy stream store; v tomto případě navíc obsahuje sériové číslo, jež jednoznačně identifikuje aplikaci, které dokument patří. Podobně jako u Macintoshe tedy neexistuje žádný problém s příponami, protože typ dokumentů se podle nich neurčuje. Epoc navíc obsahuje podporu i pro přiřazení "obyčejných souborů" s danou příponou nebo s definovaným obsahem konkrétní aplikaci, takže je možné je "importovat" prostým otevřením, stejně jako otevíráme-li dokument.

Cone a Eikon

Vyšší vrstvy aplikační podpory jsou v Epocu rozděleny na dvě úrovně. Nižší z nich, Cone, obsahuje abstraktní služby uživatelského rozhraní a grafických prvků bez vazby na jejich konkrétní implementaci. Teprve jasně oddělená vyšší vrstva, Eikon, přidává těmto prvkům konkrétní vzhled a chování. Toto rozdělení dává Epocu neobvyklou flexibilitu uživatelského rozhraní: díky společnému API Cone jsou aplikace – nebo alespoň drtivá většina jejich kódu – přenositelné mezi různými implementacemi, nabízejícími prostřednictvím různých Eikonů různá uživatelská rozhraní.

Této možnosti samozřejmě využívají malá specializovaná zařízení: zatímco kapesní počítač Ericsson MC218 má "obyčejný" Eikon, inteligentní mobil Ericsson R380 nabízí odlišné uživatelské rozhraní díky přepracovanému Eikonu. Přitom zachovává se všemi ostatními implementacemi Epocu základní kompatibilitu. Jiným příkladem využití této služby je již zmíněná (oficiálně dosud nepotvrzená) aktivita firmy Nokia, jež připravuje Eikon s uživatelským rozhraním odpovídajícím oblíbenému PalmOS.

Konverze formátů

Pro konverze nabízí Epoc nový, netradiční přístup, který bohužel, podobně jako SDK s "emulátorem" pod Windows, přináší více problémů než výhod. Epoc totiž nepublikuje formáty svých dokumentů a místo toho nabízí "engine" každé standardní aplikace jako dynamicky zaveditelnou knihovnu. Převodník "z Epocu ven" tedy nepotřebuje znát formát, protože data načte prostřednictvím služeb samotné aplikace, jež je vytvořila; další standardní služby využije pro přístup k těmto datům a na jejich základě generuje výstupní formát. Obdobně je tomu u převodníku "do Epocu": ten pomocí standardních služeb aplikace vytvoří prázdný dokument a zaplní jej daty, jež vytvoří na základě "importovaného" souboru.

Právě pro tyto konverzní programy slouží "architektura" WINC, podporovaná SDK. Epoc totiž považuje za samozřejmé, že takovéto konverze budou pracovat na stolním počítači. Tato úvaha je však hluboce chybná, a to ze dvou příčin:

- WINC podporuje jen Windows; podobně "engines" aplikací existují jen jako dynamicky zaváděné knihovny pro Windows (jen jediný "engine" textového editoru je k dispozici i ve zdrojovém tvaru; to je ovšem

málo platné, protože využívá API Epocu, a je proto přeložitelný zase jen pro WINC);

- dokonce ani pro uživatele Windows to ale nestačí: konverze formátů jsou zapotřebí často i ve chvíli, kdy žádný desktop není k dispozici – například dostaneme-li e-mail s attachmentem prostřednictvím GSM modemu.

Proto X.soft připravuje univerzální konverzní architekturu XConversion, jež pracuje přímo v Epocu a zajišťuje oboustranné konverze datových formátů podle potřeby. Na rozdíl od kompletního portabilního SDK, jež bude k dispozici v dlouhodobém výhledu, je systém XConversion těsně před dokončením. Předběžné informace o této technologii lze nalézt na www.ocs.cz/text/XSdk/WhatsXConversion.html a www.ocs.cz/text/XSdk/XConversion.html.

Ostatní

Epoc obsahuje dlouhou řadu dalších knihoven a služeb: luxusní knihovnu pro řízení a spolupráci s GSM telefony, kompletní TCP/IP a IrDA stack i služby pro sdílený přístup k databázi kontaktů a k údajům v elektronickém diáři včetně standardních prostředků pro jejich export a import (vCard). O řadě dalších služeb se pro omezený prostor ani nezmiňuji.

Některé další služby, jež Epoc sám o sobě nenabízí, budou i součástí SDK, na kterém pracuje X.soft. Již nyní jsou v rámci velmi omezené podpory XConversion připraveny například obecné beztypové kontejnery, poloautomatický garbage collector, hash tabulky nebo luxusní třída pro práci s textovými řetězci, podporující řadu standardních osmibitových kódování (včetně všech těch, která se používají pro češtinu) i Unicode.

Shrnutí

Epoc je velmi kvalitní moderní operační systém, který nabízí všechny odpovídající služby (jako je ochrana paměti nebo preemptivní multitasking). Jeho zajímavost spočívá i ve špičkové podpoře specifických vlastností mobilních zařízení, ve vysoce flexibilním uživatelském rozhraní a v široké podpoře od velkých firem (Nokia, Ericsson, Motorola, Panasonic, Sony...). Korektně podporuje nejdůležitější standardy (TCP/IP, IrDA, v budoucnosti BlueTooth...).

Stávající standardní SDK sice nabízí velmi solidní paletu knihoven s poměrně luxusními službami, je však omezeno jazykem C++ a především striktní orientací na MS Windows. Alespoň potenciálně však umožňuje vyšší portabilitu než jiná řešení, a to díky tomu, že jeho základem je plně přenositelný překladač GNU C++. Již nyní X.soft rozbíhá projekt, jehož cílem je plně portabilní SDK, podporující i daleko lepší Objective C, ale i třeba Fortran či kompilovanou (a proto rychlou) Javu.

Ze všech těchto důvodů je Epoc přinejmenším pro několik nejbližších let patrně softwarovou budoucností mobilních systémů.

Ondřej Čada