

Přehled operačních systémů reálného času (3)

Hostitelské a cílové systémy

Většina uvedených operačních systémů je velmi flexibilních, s dobrou kompatibilitou pro různé hardwarové platformy a snadnou rozšiřitelností. Většinou je k dispozici kromě vlastního RTOS také vývojové prostředí pro vývoj aplikací na osobních počítačích, které umožňuje rovněž vývoj vestavěných aplikací v prostředí operačních systémů Windows nebo Linux. Vývojové prostředí (obr. 8) zpravidla zahrnuje komplexní sadu křížových vývojových nástrojů a další vybavení včetně komunikačních nástrojů pro spojení vývojového hostitelského systému na osobním počítači s cílovým systémem pro aplikaci nejčastěji s nějakým mikroprocesorem nebo mikrokontrolérem [6].

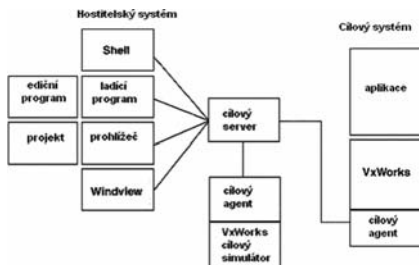
V zásadě existují dvě základní možnosti vývoje a ladění aplikací:

- Ve vlastním prostředí (self-hosted). Vývoj přímo na cílovém systému. Počítač musí mít dostatečné množství prostředků a odpovídající výkon. Běžně se využívá u počítačů typu PC.
- Křížový vývoj (cross-development). Vývoj na hostitelském systému, neboť cílový systém má omezené zdroje znesnadňující či znemožňující přímý vývoj aplikací.

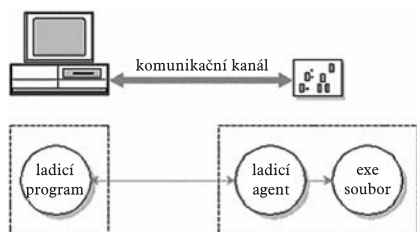
V případě ladění aplikací ve vlastním prostředí (self-hosted) ladící program (debugger) spustí ladícího (cílového) agenta a poté s ním naváže komunikační spojení.

Ladění aplikací v prostředí křížového vývoje (cross-development) je znázorněno na obr. 9. Ladící program běží na hostitelském systému a laděná aplikace na cílovém systému.

Hostitelský a cílový systém musí být při tomto způsobu vývoje a ladění vzájemně propojeny komunikačním kanálem. Jednotlivé komponenty při ladění mají různé funkce. Aplikace určuje uživatelské rozhraní a komunikaci s ladícím agentem. Ladící agent řídí laděný program. Všechny zdrojové soubory a ladící informace zůstávají na hostitelském počítači. Aby bylo možné programy ladit pomocí symbolického ladícího programu (použit zdrojové soubory), musí cílový program po kompilaci obsahovat potřebné symbolické informace, což umožňuje zadání potřebných parametrů kompilátoru.



Obr.8 Vývojové prostředí Tornado RTOS VxWorks



Obr.9 Křížové ladění v prostředí cross-development

Ladící agent může být na úrovni procesu. Při výskytu bodu přerušení (breakpoint) ladící agent převezme řízení procesu a zastaví všechna jeho vlákna. Ostatní procesy pokračují v běhu a přerušení jsou povolena. V tomto stavu je možné zjistit stav libovolného vlákna ze zastaveného procesu. Lze také vybrat vlákna, která zůstanou zastavená, a pokračovat dále. V závislosti na formě komunikačního kanálu může ladící program realizovat např. tři způsoby spojení: sériové spojení, TCP/IP spojení na určeném rozhraní, TCP/IP spojení s dynamicky přidělovaným rozhraním.

Jedním z nejmarkantnějších rozdílů při práci s vestavěnými systémy nebo jednodeskovými počítači a klasickými počítači PC je instalace operačního systému. Například na desce cílového počítače je fleš paměť, do které je nutné celý operační systém a vyvíjené aplikace zavést z hostitelského systému. Obdobně je vybavena většina jednodeskových nebo vestavěných systémů pouze, cílová paměť může být odlišného typu.

Na hostitelském systému se vytvoří obraz cílového operačního systému. Tuto operaci lze rozdělit do několika fází:

- vytvoření obrazu zaváděných souborů cílového operačního systému,
- vytvoření obrazu vestavěného souborového systému pro fleš paměť,
- sloučení obou obrazů a zavedení výsledného souboru do fleš paměti.

Pro vytvoření zaváděného obrazu operačního systému slouží speciální utilita, která pomocí určeného souboru načítá pravidla a požadavky pro vytvoření obrazu a podle nich jej generuje. Tyto řídicí soubory mají pevně danou strukturu a množinu pravidel a atributů, které lze použít.

Finální skript pro vytvoření obrazu OS, například v operačním systému QNX RTP, je následující [6]. Nejdříve se vytvoří obraz zaváděného operačního systému z řídicího souboru `sbc_geode_gxm.build`. Tento obraz se sloučí s prvotním zaváděčem (IPL – Initial Program Loader) uloženým v souboru `ipl-sbc_geode_gxm.pad`. Velikost výsledného souboru se doplní na velikost 640 KB. Obraz zaváděče je pojmenován `sbc_geode_gxm.boot`. Pak lze přistoupit k vygenerování obrazu souborového systému pomocí řídicího souboru `efs.build`. Obraz souborového systému je pojmenován `efs.image`. Sloučením obrazu zaváděče a obrazu souborového systému se získá požadovaný obraz operačního systému.

Závěr

V tomto článku nejsou jednotlivé operační systémy popsány podrobně, ale je zde uvedena jen jejich stručná charakteristika. Podrobnější popis jednotlivých systémů a jejich vzájemné porovnání by vyžadovalo podstatně rozsáhlejší publikaci. Zájemci o podrobnější informace o jednotlivých operačních systémech si mohou tyto informace zjistit na Internetu, kde jsou uvedeny firmy, které tyto RTOS dodávají a další odkazy na podrobnější informace. Některé operační systémy lze i volně stáhnout přímo z Internetu.

Článek byl zpracován v rámci řešení projektu GAČR 102/05/0467- Architektury sítí pro vestavěné systémy.

36

prof. Ing. Vilém Srovnal, CSc.

Katedra měřicí a řídicí techniky, FEI
VŠB Technická univerzita Ostrava, ČR