# Systémy reálného času (1)

**Jindřich Černohorský, Vilém Srovnal**

**Článek se zabývá výkladem základních pojmů z oblasti systémů reálného času mezi něž počítáme i náročnější systémy řízení technologických procesů, v nichž dodržení časových podmínek vyžadovaných řízenou technologií má zásadní význam nejenom pro správnou funkčnost takového systému, ale i někdy pro jeho bezpečnost, případě bezpečnost jeho okolí. Jsou diskutovány dva základní přístupy k rozvrhování procesů běžících v režimu reálného času – podle priorit a podle kritických termínů – a některé jejich varianty a vlastnosti. Teoretický aparát používaný v této oblasti je demonstrován na algoritmu RMS frekvenčně monotónního rozvrhovaní.**

## Úvod

Pro řízení reálných procesů musí být počítač naprogramován tak, aby jeho činnost byla v souladu s časovým režimem a průběžně vznikajícími požadavky řízeného systému. Říkáme, že počítač pracuje v reálném čase a celý systém nazýváme systémem reálného času, zkráceně RT systémem. Korektní chování RT systému závisí nejenom na výsledcích provedených výpočtů, ale i na čase, v němž jsou tyto výsledky vypočítány anebo na čase, kdy je proveden z počítačem provedeného vyhodnocení vypočtený akční zásah. Opožděná reakce může být zbytečná, samo zpoždění pro řízený systém někdy i nebezpečné.

U řady RT systémů jde o velmi náročné aplikace, jako jsou například řízení chemických a jaderných provozů, řízení složitých výrobních procesů, řízení dopravy, řízení letecké dopravy atp. Kritičnost uvedených aplikačních prostředí z pohledu uživatele znamená, že řídicí systémy nasazené v takových aplikacích musejí za každých, i extrémních podmínek, reagovat v čase přesně a včasně i na více současně přicházejících událostí. Zároveň musí být spolehlivé a jejich časová odezva na náhodně přicházející události musí být předvídatelná.

Předvídatelná časová odezva by měla být zajištěna i v tak obtížných případech, kdy se vyskytne současně několik událostí vyžadujících tutéž službu a tedy tytéž systémové zdroje, což znamená pro systém situaci přechodného přetížení. Jakákoliv degradace výkonu, kterou si tato situace může vynutit, musí být provedena hladce, předvídatelně a co nejsoustředěněji. Požadavek na současnost zpracování pak implikuje, že RT systém musí poskytnout možnosti paralelního resp. souběžného zpracování informací.

Bohužel problémy související s předvídatelností nelze vždy odhalit, a to ani intenzivním testováním, protože vnější podmínky, které se mohou vyskytnout v reálném provozu, nelze v testovací fázi dokonale napodobit.

Zajímavým příkladem zde může posloužit drobná softwarová závada, která byla odhalena na systému řízení střel Patriot během války v Perském zálivu [4]. Tyto střely byly určeny k ochraně Saudské Arábie proti střelám Scud s následujícím scénářem použití. Když radar zaregistruje letící objekt, vestavěný počítač vypočítá předpokládanou trajektorii a pak, aby se ujistil, že nejde o planý poplach, provede verifikaci výpočtu. Jestliže letící objekt projde specifickým (vypočteným) místem, je střela Patriot vypuštěna proti objektu, jinak je incident klasifikován jako falešný poplach.

25. února 1991 radar zaznamenal střelu Scud letící na Saudskou Arábii, palubní počítač vypočítal její dráhu, provedl verifikaci a klasifikoval událost jako falešný poplach. O několik minut později dopadla střela Scud na město Dhahram a způsobila tam ztráty na životech a obrovské ekonomické škody. Následnou analýzou bylo zjištěno, že v důsledku drobné programové chyby, kumulovaly hodiny reálného času na palubním počítači zpoždění asi 57 mikrosekund za minutu. V době incidentu byl počítač v provozu asi 100 hodin. Toto byla zcela výjimečná podmínka, se kterou do této doby nebyly žádné zkušenosti. Za tuto dobu nakumuloval počítač celkem 343 milisekund zpoždění. To způsobilo při výpočtu pozice na kalkulované dráze odchylku 687 metrů. Chyba byla opravena druhý den po události.

V závislosti na přísnosti časových omezení klasifikujeme RT systémy jako, měkké (soft) nebo kritické (hard). Zatímco prodlužující se odezva u soft systémů může být tolerována za cenu nárůstu nákladů spojených s provozem systému, v případě kritických systémů, nemůže být zpoždění tolerováno vůbec.

Vztah RT systému k vnějšímu prostředí může být popsán na základě událostí, na něž musí systém reagovat prostřednictvím nějakých akcí realizovaných jako úlohy resp. programy. Jeden typický druh akcí je definován na časovém základě. Jde o periodicky se opakující akce nebo akce prováděné v určitém čase. Hovoříme proto o úlohách řízených časem. Druhý typ akcí vychází z událostí, které nastávají v řízeném systému a jsou signalizovány z vnějšku například pomocí přerušení anebo z řídicího systému dalšími výpočetními procesy vyhodnocujícími situaci na základě měření technologických parametrů a jeho následného zpracování. Hovoříme pak o úlohách řízených událostmi. Některé akce mohou být také vyvolány na základě komunikace s obsluhou a hovoříme o interaktivně řízených úlohách.

Lze tedy shrnout:

O systémech reálného času hovoříme tehdy, jestliže
- Pořadí výpočtů je určeno tokem času nebo událostmi externími vzhledem k počítači
- Výsledky jednotlivých výpočtů mohou záviset na hodnotě proměnné „čas“ v době provádění výpočtu nebo když se hodnota okamžitého času bere do úvahy jako jeden z parametrů výpočtu
- Korektní chování RT systému závisí nejenom na výsledcích výpočtů provedených procesem, ale i na čase, v němž jsou tyto výsledky vypočítány
- Opožděná reakce může být zbytečná, samo zpoždění v některých případech i dokonce nebezpečné

## Paralelismus a RT systémy

Systémy reálného času musí mít programovou podporou pro souběžný, pseudoparalelní chod více úloh. To na jednoprocesorovém počítači umožňuje víceúlohový (multitasking) operační systém anebo alespoň jeho tzv. jádro (kernel), které umožňuje řídit souběžné provádění více programů. Říkáme, že jádro přepíná řízení mezi jednotlivými procesy, tj. postupně podle zvolené strategie přiděluje CPU jednotlivým procesům. Protože každý z relativně autonomně běžících procesů má svůj logický kontext – stav výpočtu kterého dosáhl v daném okamžiku – říkáme též operaci předávání procesoru mezi jednotlivými procesy přepínání kontextu.

Pokud jsou řízeny operačním systémem, nazývají se souběžně prováděné programy procesy, úlohy nebo tasky. Souběžnost lze často také realizovat na úrovni jednoho programu jeho rozčleněním do souběžně provozovatelných částí, které se nazývají vlákna (thread). Programově jsou vlákna popsána jako procedury a v principu jsou založena na stejných a řízena pomocí stejných mechanismů jako procesy.

Na první pohled se může zdát, že realizovat na jednoprocesorovém počítači jakýsi virtuální paralelismus je spíše komplikací a že nemůže přinést žádný reálný efekt, avšak opak je pravdou. Efekt tohoto řešení spočívá v tom, že

- Systém je možno rozdělit na programové části (procesy, nebo vlákna), z nichž každá zodpovídá za provedení určité programové akce vyvolané nějakou událostí nebo na základě času
- Jednotlivé procesy mohou mít přiřazeny priority, čímž může být vyjádřena důležitost jejich reakce v systému a mohou při zpracování dostat přednost před procesy s nižší prioritou, pokud se vyskytne potřeba reagovat v témže čase na více událostí
- Dojde-li při zpracování nějakého procesu k jeho selhání, může být ukončen jen tento proces a ostatní procesy mohou pokračovat ve zpracování
- Jednoznačný vztah mezi reálnými procesy řízeného systému a mezi procesy „pověřenými“ řízením nebo modelováním těchto reálných procesů umožňuje snadněji a méně komplikovaněji naprogramovat celý systém

Jedním ze zásadních problémů víceúlohových systémů, resp. RT systémů, je proto způsob rozvrhování a řízení jejich procesů v čase.

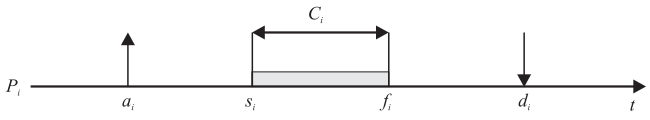
## Časová omezení

Kritickým časovým omezením vztaženým k procesu je „deadline“ (časová uzávěra, kritický termín, termín), který představuje čas, před nímž by měl proces ukončit své provedení. Kvantifikace časových omezení pro kritické RT systémy se zpravidla určuje na základě výpočtů vycházejících z fyzikálních zákonů popisujících řízený proces.

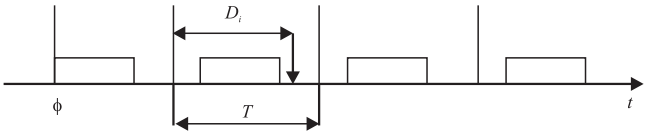
Ve vztahu k důsledkům, které může mít promeškání kritického termínu se RT úlohy, procesy, dělí do dvou tříd, „hard“ a „soft“. Proces P má charakter hard (kritický, nepoddajný), jestliže jeho dokončení po termínu může mít pro systém katastrofální následky. Proces je typu soft (měkký, poddajný), jestliže nesplnění jeho kritického termínu snižuje výkon systému, ale neohrožuje jeho korektní chování.

Obecně může být proces $P_i$ charakterizován řadou parametrů, jako jsou například:

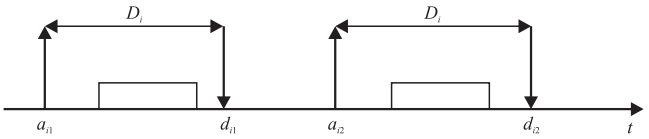
- Čas příchodu (arrival time) $a_i$ – čas, kdy se proces stává připravený pro provedení.
- Výpočetní čas (computation time) $C_i$ – čas provední procesu, tj. čas potřebný pro procesor, aby mohl proces provést bez přerušení
- Kritický termín (deadline) $d_i$ – čas, před nímž by měl být proces dokončen, aby se zabránilo škodám na systému.
- Čas spuštění (start time) $s_i$ – čas, v němž se proces začne provádět.
- Čas ukončení (finishing time) $f_i$ – čas, v němž proces dokončí své provedení.
- Kritičnost (criticalness) – parametr popisující důsledky nesplnění termínu
- Opoždění (lateness) $L_i$ – čas $L_i = f_i - d_i$ a vyjadřuje zpoždění dokončení procesu vzhledem k termínu. Skončí-li proces před termínem je $L_i < 0$.




**Obr.1 Charakteristické parametry procesu $P_i$**




**Obr.2 Periodický proces**




**Obr.3 Aperiodický proces**

- Čas volnosti (laxity, slack time) $X_i - X_i = d_i - a_i - C_i$ je maximální čas, kdy může být úloha odložena při své aktivaci, aby se ještě dokončila před termínem

Jiná časová charakteristika procesu se týká pravidelnosti jeho aktivace. Procesy mohou být definovány jako periodické nebo aperiodické.

Periodické procesy jsou tvořeny nekonečnou posloupností identických aktivit i nazývaných instance nebo joby, které jsou aktivovány s pravidelnou frekvencí.

Aktivační čas první periodické instance se nazývá fáze. Je-li $\phi_i$ i fáze periodického procesu $\tau_i$, aktivační čas $k$-té instance je dán $\phi_i + (k-1)T_i$, kde $T_i$ se nazývá perioda procesu. V řadě praktických příkladů může být periodický proces úplně charakterizován výpočetním časem $C_i$ a jeho relativním kritickým termínem $D_i$, který se často pokládá za koincidentní s koncem periody. Navíc parametry $C_i$, $D_i$ a $T_i$ se považují za konstantní pro každou instanci. Aperiodické procesy jsou též tvořeny nekonečnou sekvencí identických aktivit, ale jejich aktivace nejsou pravidelné.

## Literatura


[1] HALANG, W. A., STOYENKO A. D.: Real Time Computing, Springer-Verlag, 1994, ISBN 0-387-57558.

[2] HALANG, A. W., STOYENKO, A. D.: Constructing predictable real time systéme, Kluwer Academic Publishers, Boston/ Dordrecht/London, 1997

[3] LIU, C. L., LAYLAND, J. W.: Scheduling algorithms for multiprogramming in a hard-real-time environment, Journal of the Association for Computing Machinery, 20 (1), 1973

[4] BUTTAZO, G. C.: Hard real-time computing systems, predictable scheduling algorithms and application, Kluwer Academic Publishers, 1998

[5] ČERNOHORSKÝ, J., GARZINA, R.: Algoritmy rozvrhování v úlohách reálného času. Automatizace 42 (1999), č. 5, s. 318 – 324

[6] DIBBLE, P. C.: Real-Time JAVA Platform Programming, The Sun Microsystem press, 2002, ISBN 0-13-028261-8


*Pokračovanie v budúcom čísle.*


**doc. RNDr. Jindřich Černohorský, CSc.**
**prof. Ing. Vilém Srovnal, CSc.**

**Katedra měřicí a řídicí techniky, FEI**
**VŠB Technická univerzita Ostrava**
**17. listopadu 15/2172**
**708 33 Ostrava-Poruba, ČR**
**e-mail: jindrich.cernohorsky@vsb.cz**
**vilem.srovnal@vsb.cz**