

AUTOMA

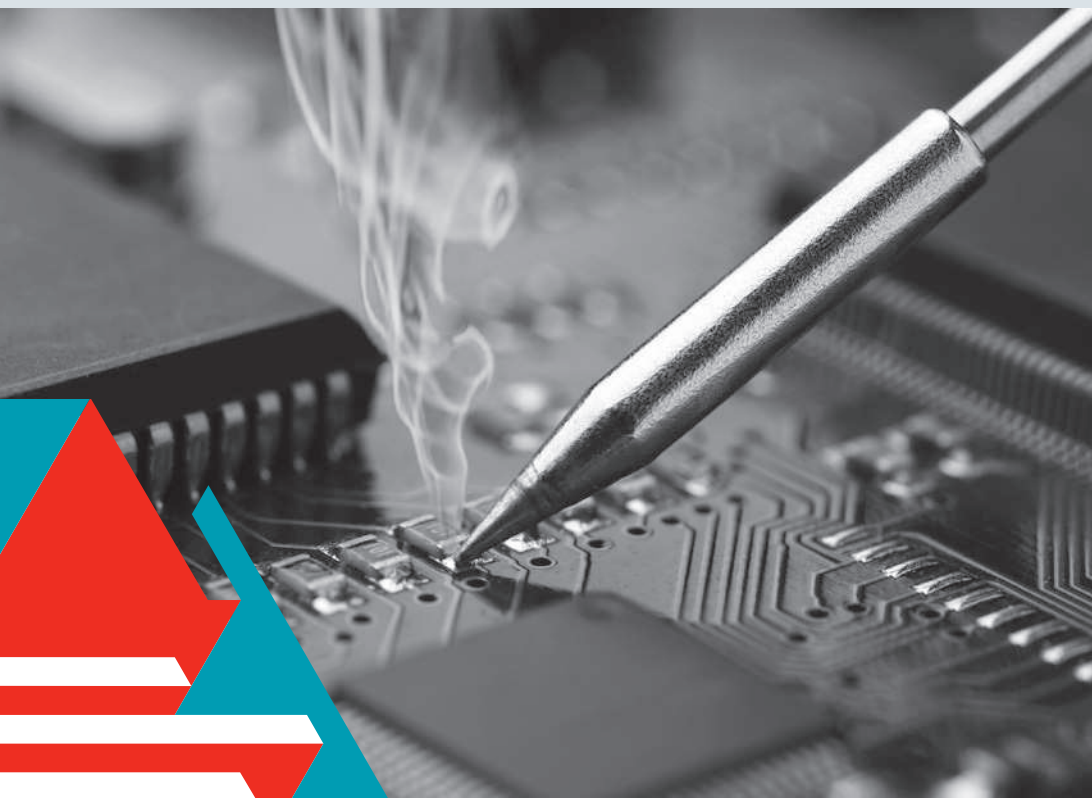
časopis pro automatizační techniku



www.automa.cz

Cena 52 Kč

12 PROSINEC 2012



Zabezpečení řídicích,
informačních
a komunikačních
systémů v průmyslové
praxi

Virus Duqu - skrytá
hrozba

Integrace
bezpečnostních systémů
v budovách

Elektromechanické
bezpečnostní spínače
a zámky

Rizika vzniku
nebezpečné atmosféry
v elektrické peci
a opatření k jejich
eliminaci

Plánování úloh
v systémech RT:
přetížení systému



**DISTRELEC:
TOP-KVALITA V SRDCCI EVROPY.**

Už přes 40 let je součástí skupiny Dätwyler.

A Datwyler Company

 **DISTRELEC**



Plánování úloh v systémech RT - III: přetížení systému

V dosavadních člancích na téma návrhu časově kritických systémů a plánování úloh tzv. reálného času (*real-time*, RT, úlohy RT) uveřejněných v časopise *Automa* se předpokládá, že výpočetní požadavky úloh během dané doby v souhrnu nepřesáhnou výpočetní možnosti procesoru. Toto nelze vždy zaručit. V tomto článku je pojednáno o mechanismech zajišťujících předvídatelné chování systému i při přetížení procesoru.

V předcházejících člancích [4] až [9] bylo předpokládáno, že v rámci předem stanoveného intervalu součet výpočetních požadavků úloh nikdy nepřesáhne výpočetní možnosti procesoru, na kterém úlohy mají běžet (tj. bylo předpokládáno, že $U \leq 1$; symbolika použitá v článku je zavedena v [5]). Proto musel být systém navržen způsobem garantujícím jeho provozuschopnost i v nejhorsím uvažovaném případě stavu a chování jeho okolí. Nicméně, jestliže by tento nejhorší možný případ nebyl či nemohl být stanoven s dostatečnou přesností, systém by s velkou pravděpodobností nebyl schopen na vzniklé situace reagovat očekávaným způsobem a mohl by se např. náhle a nekontrolovatelně zhroutit.

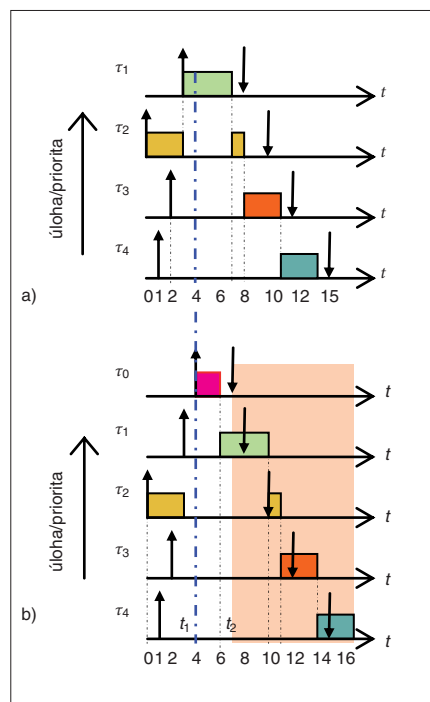
Naznačené nepříznivé situace mohou nastat jako důsledek mnoha zcela reálných příčin, mezi které patří např. změna v prostředí, neočekávané zpoždění signálu v komunikačním kanálu, přechodná chyba v systému či nadměrná stimulace systému asynchronními podněty. Tento článek bude zaměřen na přehled mechanismů garantujících předvídatelné chování systému i v situacích, kdy je procesor přetížen.

Dominový efekt

Přetížení ukazuje *obr. 1*, kde v části a) je ilustrován mechanismus přidělování úloh EDF (*Earlier Deadline First*), v němž jsou všechna časová omezení úloh dodržena. Avšak je-li v „nevhodném“ čase t_1 do systému zavedena nová úloha τ_0 s „nevhodně vysokou“ prioritou, její provádění může vést k tzv. *dominovému efektu*, ukázanému v části b) obrázku – je zřejmé, že časová omezení všech zbylých úloh v systému budou překročena. Podobně jako EDF ani ostatní mechanismy jako RM (*Rate Monotonic*), DM (*Deadline Monotonic*) či LLF (*Least Laxity First*; [6]) nejsou na přetížení „připraveny“, a dojde-li k němu, neplní svůj původní účel a namísto toho selhávají.

I z uvedeného jednoduchého příkladu vyplývají dva naprosto klíčové důsledky. První lze shrnout následovně: požadují-li úlohy více výpočetní doby, než kolik je jí k dispozici (tj. je-li systém přetížen), nelze všechny úlohy provést, a tedy ani dodržet časo-

vá omezení všech úloh v systému. Z tohoto důsledku plyne další, neméně závažný: nemá-li se v důsledku přetížení systém náhle a nekontrolovatelně zhroutit, musí být rozhodnuto, které instance úloh na procesoru dále poběží (a dodrží svá časová omezení), které poběží až poté, co přetížení pomine (tj. poskytnou odezvu se zpožděním), a které nepoběží vůbec (tj. neposkytnou žádnou odezvu).



Obr. 1. Ilustrace dominového efektu vzniklého v důsledku přetížení procesoru

Zatížení a přetížení

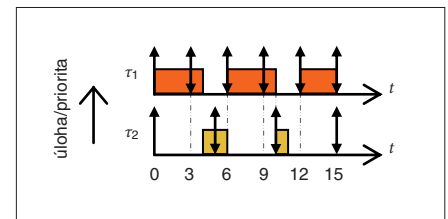
K představení principů souvisejících s výchozími pojmy vztahující se k tomuto tématu.

Jako první vymezme *zatížení procesoru* ρ , dávající do poměru dobu C_i činnosti procesoru požadovanou úlohami τ_i k jejich běhu a dobu D_i , která je na procesoru k dispozici pro včasné dokončení úloh ($i = 1, \dots, n$). V nejjednodušším případě, tj. pro množinu nezávislých, periodických, preemptivně plá-

novaných úloh, pro jejichž parametry D_i , T_i platí $D_i = T_i$, se ρ rovná U , tj. platí [5]

$$\rho = U = \sum_{i=1}^n \frac{C_i}{T_i} \quad (1)$$

Pro obecnou množinu úloh však bude ρ proměnné v čase a bude závislé zejména



Obr. 2. Ilustrace prevence přetížení při použití několikařádkového modelu úloh

na na časech příchodů r_i jednotlivých úloh a vzdálenosti absolutních časových mezí odezvy d_i od aktuálního času t .

Statické zatížení procesoru ρ_{st} úlohou τ_i lze určit jako

$$\rho_{st} = \frac{C_i}{D_i} = \frac{C_i}{d_i - r_i} \quad (2)$$

Tedy při $C_i = D_i = d_i - r_i$ musí být úloze τ_i přidělen procesor; jinak nebude dokončena včas, tj. do d_i . Z pohledu analýzy ρ v čase je pak s každým přepnutím kontextu, vyvolaným např. příchozem nové úlohy do systému, vhodnější analyzovat celkové okamžité zatížení procesoru v intervalu $[t, d_{max}]$, kde d_{max} je největší z hodnot d_i úloh τ_1, \dots, τ_m připravených k běhu v čase t . Jelikož v tomto intervalu jsou obsaženy i meze d_1, \dots, d_m všech připravených úloh, je třeba analyzovat každý z podintervalů $[t, d_1], \dots, [t, d_m]$ intervalu $[t, d_{max}]$ za účelem určit, zda do času d_i úlohy τ_i stihne být včas dokončena i každá úloha τ_k s $d_k \leq d_i$.

Okamžité zatížení procesoru v intervalu $[t, d_i]$ je pak dáno vztahem

$$\rho_i(t) = \frac{\sum_{\tau_k: d_k \leq d_i} C_k(t)}{D_i(t)} \quad (3)$$

kde

$C_k(t)$ je doba zbývající k dokončení úlohy τ_k , $D_i(t) = d_i - t$ doba zbývající do d_i v čase t . Okamžité zatížení procesoru v čase t je $\rho(t)$ a rovná se největšímu zatížení procesoru v uvedených časových intervalech podle vztahu

$$\rho(t) = \max_{i=1, \dots, m} [\rho_i(t)] \quad (4)$$

Platí-li $\rho(t) > 1$, procesor je v čase t přetížen. Postup výpočtu $\rho(4)$ pro plány z obr. 1a, b je ilustrován v tab. 1.

Východiska z přetížení

V praxi existují dva základní způsoby, jak přetížení čelit. První způsob představují mechanismy založené na manipulaci s parametry úloh [2]. Jejich činnost spočívá v dočasné změně hodnot parametrů vybraných úloh za účelem předejít přetížení procesoru nebo zkrátit dobu jeho trvání (obvykle řízeným zmenšováním výpočetních požadavků úloh, prodlužováním dob D , T apod.). Jde o mechanismy vhodné v úlohách (např. multimediálních), u nichž je přípustná dočasná degradace výkonu systému. Alternativou jsou mechanismy plánování založené na důležitosti (významnosti) úloh [1], kdy při přetížení jsou nedůležité úlohy z plánu vyřazeny, zatímco důležité úlohy jsou, s původními parametry, v plánu ponechány k včasnému dokončení.

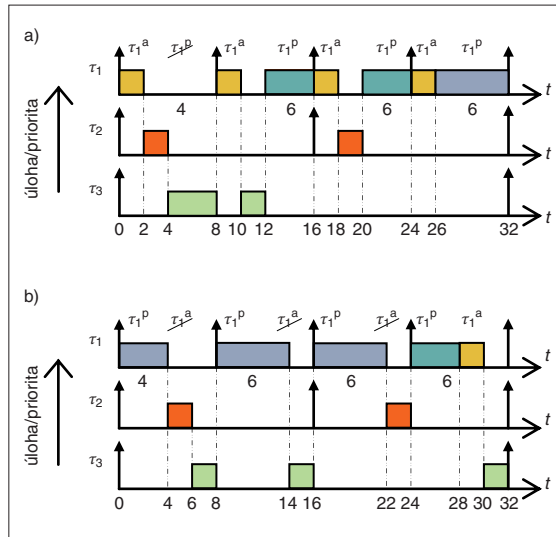
Zamezení přetížení řízenou degradací výkonu systému

Z mechanismů zamezujících přetížení procesoru řízenou degradací výkonu systému jsou významné techniky několikarámcového modelu, variantní instance úloh, iterační technika a technika pružného modelu úloh, popsány dále.

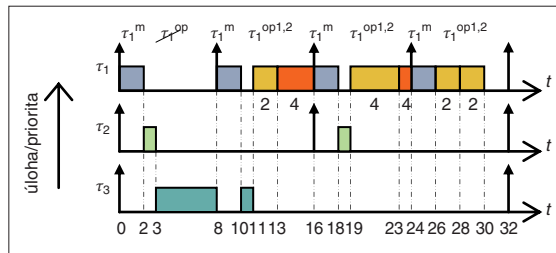
Několikarámcový model úloh

Předejít přetížení redukcí špiček zatížení procesoru lze při použití přístupu označo-

vaného jako *několikarámcový (multi-frame) model*. Tento přístup umožňuje stanovit pro každou instanci periodické úlohy jinou hodnotu parametru C . V rámci modelu úloh je pak namísto jedné hodnoty C použita uspořádaná n -tice hodnot parametru modelující n po sobě následujících instancí téže úlohy



Obr. 3. Ilustrace k technice s (a) přednostním a (b) odloženým prováděním τ_i^a



Obr. 4. Ilustrace k technice vhodné pro iterační výpočty

s dobami běhu danými touto n -ticí. Daný přístup je ilustrován na obr. 2, zobrazujícím plán RM tvořený několikarámcovou úlohou $\tau_1(r; C; D = T) = \tau_1(0; 3, 1; 3)$ a „běžnou“ jedno-

rámcovou úlohou $\tau_2(r, C, D = T) = \tau_2(0, 1, 5)$, kde liché (sudé) instance τ_1 běží s hodnotou $C = 3$ ($C = 1$), čímž je dosaženo $\rho = 1$. Kdyby totiž každá instance τ_1 běžela s hodnotou $C = 3$, jistě by platilo $\rho > 1$ a některá z úloh by nebyla dokončena včas.

Variantní instance úloh

Z obdobného principu vycházejí i další přístupy, např. přístup založený na *variantních instancích úloh*. Ten předpokládá, že pro každou úlohu τ_i jsou realizovány dvě varianty – *primární* τ_i^p , produkující odezvu v požadované vysoké kvalitě, avšak za relativně dlouhou, mnohdy také velmi proměnnou dobu, a *alternativní* τ_i^a , produkující odezvu v přijatelně nízké kvalitě, nicméně za relativně krátkou, ideálně deterministickou dobu. Plánovač musí být navržen tak, aby bylo vždy zaručeno včasné provedení alespoň jedné z variant τ_i^a , τ_i^p , přičemž je žádoucí, aby τ_i^p byla dokončována co nejčastěji. Tento přístup může být realizován dvěma technikami – technikou s *přednostním* prováděním τ_i^a a technikou s *odloženým* prováděním τ_i^a . Princip první z technik je takovýto:

- τ_i^a vždy běží před τ_i^p ,
 - τ_i^p je spuštěna až tehdy, je-li dostatek času k jejímu včasnému provedení,
 - je-li τ_i^p dokončena včas, výsledek τ_i^p je použit místo výsledku τ_i^a .
- Druhá z technik funguje takto:
- τ_i^p vždy běží před τ_i^a ,
 - není-li již dostatek času k včasnému provedení τ_i^p , je τ_i^p ukončena a je spuštěna τ_i^a , a to s předstihem nutným k včasnému dokončení τ_i^a i ostatních úloh.

Je-li hodnotícím kritériem kvalita služby vztahená k vytížení procesoru, lze za lepší označit techniku s odloženým prováděním τ_i^a , jejíž přednost spočívá v tom, že τ_i^a je spuštěna pouze tehdy, je-li zřejmé, že τ_i^p nebude možné dokončit včas. To však vyžaduje zvláštní režii na straně plánovače spojenou zejména se sledováním včasného provádění τ_i^p a popř. ukončením τ_i^p a spuštěním τ_i^a . Obě techniky jsou ilustrovány na obr. 3 při použití následující množiny úloh $\tau_i(r_i, C_i, D_i = T_i)$: $\tau_1^a(0, 2, 8)$, $\tau_1^p(0; 4, 6, 6; 8)$, $\tau_2(0, 2, 16)$, $\tau_3(0, 6, 32)$. Instance τ_1 dokončené (nedokončené) včas jsou vyznačeny okrově/modře (modrozeleně).

Iterační technika

Obdobný přístup existuje pro úlohy zpracovávající iterační výpočty. Je předpokládáno, že existují úlohy, jejichž provádění je možné rozčlenit na *povinnou část* τ_i^m (*mandatory*), jejímž úkolem je zahájit výpočet a včas produkovat výsledek zatížený přijatelnou chybou, a na *volitelné části* τ_i^{op} (*optional*), z nichž každá zpřesňuje výsledky produkované předcházejícími instancemi τ_i – nepřesnost výsledku tedy klesá s rostoucím počtem dokončených instancí τ_i^{op} . K ilustraci

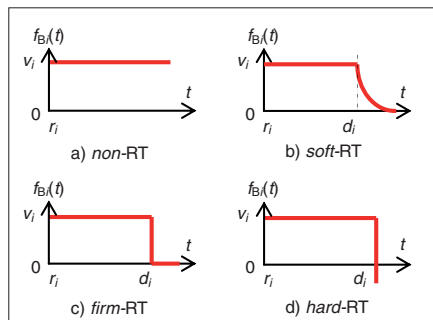
Tab. 1. Ilustrace výpočtu $\rho(t)$ pro $t = 4$ a plány z obr. 1

$t = 4$	Příklad vyhodnocení $\rho(t)$ pro situaci z obr. 1	
	a)	b)
$\rho_0(t)$	0	$\frac{C_0(t)}{D_0(t)} = \frac{2}{d_0 - t} = \frac{2}{7 - 4} = \frac{2}{3}$
$\rho_1(t)$	$\frac{C_1(t)}{D_1(t)} = \frac{3}{d_1 - t} = \frac{3}{8 - 4} = \frac{3}{4}$	$\frac{C_0(t) + C_1(t)}{D_1(t)} = \frac{2 + 4}{d_1 - t} = \frac{6}{4}$
$\rho_2(t)$	$\frac{C_1(t) + C_2(t)}{D_2(t)} = \frac{3 + 1}{d_2 - t} = \frac{3 + 1}{10 - 4} = \frac{4}{6}$	$\dots = \frac{2 + 4 + 1}{10 - 4} = \frac{7}{6}$
$\rho_3(t)$	$\dots = \frac{3 + 1 + 3}{12 - 4} = \frac{7}{8}$	$\dots = \frac{2 + 4 + 1 + 3}{12 - 4} = \frac{10}{8}$
$\rho_4(t)$	$\dots = \frac{3 + 1 + 3 + 3}{15 - 4} = \frac{10}{11}$	$\dots = \frac{2 + 4 + 1 + 3 + 3}{15 - 4} = \frac{13}{11}$
$\rho(t) = \max_{i=0, \dots, 4} [\rho_i(t)]$	$\frac{10}{11} \approx 90,91\%$	$\frac{6}{4} \approx 150\%$
Přetížení detekováno?	NE [$\rho(4) = \frac{10}{11} < 1$]	ANO [$\rho(4) = \frac{6}{4} > 1$]

na obr. 4 byla použita následující množina úloh $\tau_i(r_i, C_i, D_i = T_i)$: $\tau_1^m(0, 2, 8)$, $\tau_1^{op1}(0; 2, 2, 4, 2; 8)$, $\tau_1^{op2}(0; 2, 4, 4, 2; 8)$, $\tau_2(0, 1, 16)$, $\tau_3(0, 6, 32)$. Instance τ_1 dokončené (nedokončené) včas jsou vyznačeny zeleně (oranžově).

Pružný model úloh

Další přístup, nazvaný pružný (elastic) model, připouští, že hodnoty period úloh se mohou, v rámci předem vymezeného rozsahu, měnit. Nastane-li např. situace, že některá z úloh je vyvolána dříve, než bylo předpokládáno (tj. perioda úlohy je zkrá-



Obr. 5. Ilustrace k tvarům funkcí zisku

cena), dočasně naroste činitel využití procesoru touto úlohou, což může vést k přetížení. Ostatní elastické úlohy však mohou na tento nárůst zareagovat vhodným dočasným prodloužením svých period za účelem zmenšit jejich činitele využití procesoru a udržet systém v nepřetíženém stavu. Tento přístup ilustrujeme při použití množiny úloh $\tau_i(r, C, D = T)$: $\tau_1(0, 10, 20)$, $\tau_2(0, 10, 40)$, $\tau_3(0, 15, 70)$ a mechanismu EDF. Platí $U = 10/20 + 10/40 + 15/70 = 0,964$. Jelikož $U < 1$, systém není přetížen. Kdyby se však změnila hodnota T_3 ze 70 na 50, platilo by $U = 10/20 + 10/40 + 15/50 = 1,05$, což by již vedlo k přetížení procesoru. Této situaci lze však předejít např. tím, že na zmíněný pokles T_3 bude reagováno zvětšením T_1 na 22 a T_2 na 45, což povede na $U = 10/22 + 10/45 + 15/50 = 0,997$, tedy k potlačení přetížení.

Zamezení přetížení zavedením důležitosti úloh

Druhé východisko z přetížení představují mechanismy založené na tzv. důležitosti (value/importance) úloh. Aby nenastal domino-

vý efekt, jsou při $\rho > 1$ z plánu vyřazovány úlohy s nejmenší důležitostí.

Důležitost úloh

Hodnotu důležitosti v_i lze úloze τ_i přiřadit různým způsobem. Obvykle se však volí jedna z těchto možností:

- a) v_i je rovna konstantě V_i nezávislé na parametrech τ_i ,
- b) v_i je funkcí parametrů τ_i , např. C_i ,
- c) v_i je dána kombinací ad a), b), např. podílem $V_i/C_i(t)$.

Plánovač tedy rozhoduje nejen na základě „běžných“ – např. RM, DM či EDF – priorit zohledňujících časovou kritičnost úloh určenou v podstatě hodnotou D_i , ale také na základě hodnot v_i , tj. důležitosti úlohy pro systém jako celek. Zdůrazňme, že zatímco časovou kritičnost úlohy lze strojově stanovit na základě popisu chování systému, důležitost úlohy pro chod systému takto vyhodnotit nelze a přinejmenším je nutné ji konzultovat s návrhářem systému.

Pro ilustraci důležitosti uveďme systém, v němž běží úloha τ_1 , určená k překreslování informačního displeje s periodou $T_1 = 25$ ms, a úloha τ_2 , určená ke vzorkování teploty a tlaku v plynovém kotli s periodou $T_2 = 100$ ms. Zatímco časová kritičnost úloh τ_1, τ_2 je jednoznačně určena zvoleným mechanismem přiřazování priorit a např. v případě RM vede na $P_1 > P_2$, z pohledu řízení je jistě důležitější úloha τ_2 – ta totiž, na rozdíl od τ_1 , neslouží k „pouhému“ zobrazování informací uživateli např. za účelem dosahnout většího komfortu a lepší informovanosti obsluhy, ale má přímý vliv na řízení kotle.

Funkce zisku

V souvislosti s mechanismy založenými na důležitosti úloh je nutné zmínit tzv. funkci přínosu/zisku (benefit/utility function) $f_{B_i}(t)$, přiřazující s ohledem na plán v čase t úloze τ_i hodnotu indikující, jakým přínosem pro systém bude zařazení τ_i do plánu. Tvar funkce f_{B_i} vyplývá jednak z typu úlohy (tj. non-RT, soft-RT, firm-RT, hard-RT) a jednak ze specifikace jejího chování. Oborem hodnot funkce f_{B_i} je podmnožina intervalu $(-\infty, v_i]$. Návrhová hodnota funkce f_{B_i} je určena k ohodnocení vlivu dokončení τ_i na systém v čase t (obr. 5). V případě systémů RT pro každou τ_i dokončenou včas v čase t jistě platí $f_{B_i}(t) > 0$, tj. včasná dokončení úloh vždy zname-

nají přínos pro systém (obr. 5, t v rozmezí r_i až d_i). Avšak pro odezvy poskytnuté v čase $t > d_i$ již pro přínos, v závislosti na typu úlohy, platí $f_{B_i}(t) \leq 0$. V případě na obr. 5b např. přínos klesá k 0 spojitě podle exponenciální závislosti (tedy odezva může být pro systém přínosná i „nějakou dobu“ po překročení d_i), v případě obr. 5c přínos klesá k 0 skokově (tj. odezva je po překročení d_i pro systém bezcenná) a v případě podle obr. 5d klesá opět skokově, ale k $-\infty$ (tj. po překročení d_i může být odezva pro systém a jeho okolí nejen bezcenná, ale dokonce nežádoucí či riziková).

Pro vyhodnocení celkového přínosu pro systém určíme pro systém tvořený úlohami z $\Gamma = \{\tau_1, \dots, \tau_n\}$ s hodnotami důležitosti v_1, \dots, v_n a plánovací mechanismus M funkci

$$\Gamma_M(t) = \sum_{i=1}^n f_{B_i}(t) \quad (5)$$

vrazející zisk dosažený v čase t při plánování úloh z Γ mechanismem M. Dále, nezávisle na M, stanovme

$$\Gamma_{\max} = \sum_{i=1}^n v_i \quad (6)$$

tj. největší možný zisk dosažitelný při včasném dokončení všech úloh z Γ . Je-li v čase t systém přetížen, alespoň jedna z úloh nestihne být dokončena včas, a tedy $\Gamma_M(t) < \Gamma_{\max}$, tzn. při přetížení nelze největší hodnoty zisku (Γ_{\max}) dosáhnout. Princip mechanismů založených na důležitosti úloh pak spočívá ve snaze co nejvíce přiblížit hodnotu $\Gamma_M(t)$ hodnotě Γ_{\max} , tj. maximalizovat zisk systému s ohledem na aktuální hodnotu přetížení.

Kategorie mechanismů založených na důležitosti úloh

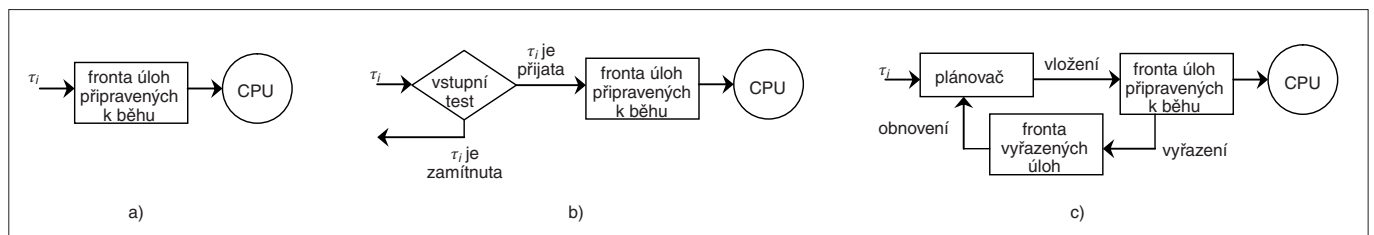
Mechanismy založené na důležitosti úloh lze klasifikovat do těchto kategorií:

- mechanismy s nejlepší snahou,
- mechanismy založené na vstupním testu přijetí,
- robustní plánovací mechanismy.

Detaily ke každé z kategorií jsou uvedeny v dalším textu.

Mechanismy s nejlepší snahou

Mechanismy s nejlepší snahou (best-effort scheduling) se vyznačují tím, že každá nově příchozí úloha je zařazena do fronty úloh připravených k běhu i tehdy, je-li systém přetížen (obr. 6a). Procesor je



Obr. 6. Ilustrace k mechanismům založeným na důležitosti úloh (CPU – procesor)

úlohám přiřazován způsobem zaručujícím přednostní provádění úloh s většími hodnotami důležitosti v_i . Méně důležité úlohy z tohoto pohledu sice „hladovějí“, avšak možnost přidělit jim procesor není zcela vyloučena. Mezi mechanismy této kategorie patří např. LBESA (*Locke's Best Effort Scheduling Algorithm*) či DASA (*Dependent Activity Scheduling Algorithm*) a lze se s nimi setkat zejména při realizaci kritických úloh – jmenujme např. realizaci systému včasného varování AWACS.

Princip mechanismu DASA lze zjednodušit následovně. Při každém volání plánovače je vyprázdněna množina úloh určených k zařazení do nově připravované fáze plánu; poté jsou úlohy seřazeny sestupně podle hodnot tzv. *hustoty důležitosti* (*value density*) $V_{D_i}(t)$, stanovené pro úlohu τ_i a čas t jako $V_{D_i}(t) = V_i/C_i(t)$. Smyslem parametru $V_{D_i}(t)$ je to, aby s ohledem na důležitost úlohy V_i ohodnotil „návrátost“ času již investovaného do běhu úlohy τ_i . Pro neměnnou hodnotu V_i tedy $V_{D_i}(t)$ roste s časem, který úloha již strávila v procesoru (tj. nepřímou úměrnost době $C_i(t)$ potřebné k dokončení úlohy). V pořadí klesajících hodnot $V_{D_i}(t)$ jsou poté po řadě prováděny tyto činnosti:

1. Úloha je umístěna do množiny úloh určených k zařazení do plánu.
2. Je analyzováno, zda vytvoření plánu složeného z těchto úloh nepovede k přetížení; došlo-li by k němu, je posledně vložená úloha z množiny vyjmuta.

Procesor je pak přidělen úloze s nejbližší absolutní časovou mezí d .

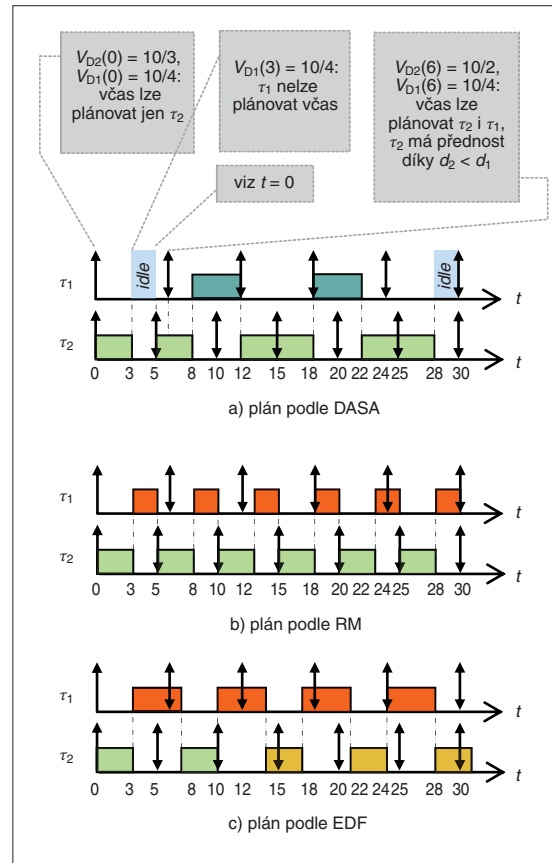
Princip mechanismu DASA je, spolu s porovnáním s RM a EDF, ilustrován na obr. 7. V ilustraci jsou předpokládány úlohy τ_1, τ_2 s následujícími parametry: $C_1 = 4, T_1 = 6, C_2 = 3, T_2 = 5, V_1 = V_2 = 10$. Zeleňou (červenou) barvou jsou označeny instance úloh (ne)dokončených včas. Je zřejmé, že mechanismus DASA je schopen čelit přetížení podstatně lépe než konvenční mechanismy RM a EDF. Za cenu „obětování“ vybraných instancí méně důležitých úloh se zmenší výpočetní požadavky kladebné na procesor – v příkladu na obr. 7 bude procesor v modře zbarvených intervalech dokonce nečinný (*idle*).

Mechanismy založené na testu přijetí

Pro mechanismy založené na vstupním testu přijetí (*admission/guarantee test*) je charakteristické, že každá nově přichází úloha prochází vstupním testem zaručujícím $\rho \leq 1$. Je-li tento test úspěšný, je úloha přijata k za-

řazení do fronty úloh připravených k běhu. Jinak je úloha zamítnuta a systémem není dále zpracovávána, tj. možnost přidělit jí procesor zcela zaniká.

K mechanismům z této kategorie patří např. algoritmus D_{over} , založený na následu-



Obr. 7. Ilustrace k mechanismu DASA (a) doplněná o porovnání s mechanismy RM (b) a EDF (c)

řazením principu: při detekci přetížení je důležitost V_τ přichází úlohy τ porovnávána s důležitostí V_{pre} úloh přerušovaných preemptivně a důležitostí V_{act} aktuálně běžících úloh. Platí-li

$$V_\tau > (1 + \sqrt{K})(V_{pre} + V_{act}) \quad (7)$$

kde K je poměr největší k nejmenší z hodnot důležitosti na dané množině úloh, je τ přijata, jinak je zamítnuta.

Robustní mechanismy

Typickým rysem kategorie tzv. *robustních plánovacích mechanismů* (*robust scheduling*) je existence této trojice pravidel pro práci s úlohami:

- pravidla A pro vložení úlohy do fronty úloh připravených k běhu (kritériem může být např. hodnota parametru d úlohy),
- pravidla B pro vyřazení úlohy z fronty úloh připravených k běhu při detekci přetížení (zde je kritériem hodnota důležitosti úloh); vyřazené úlohy jsou uchovávány ve frontě vyřazených úloh,
- pravidla C pro obnovení možnosti úloh být znovu předmětem plánování.

Do této kategorie patří např. mechanismus RED (*Robust Earliest Deadline First*). Jeho princip je následující. Pravidlo A je realizováno mechanismem EDF, pravidlo B při detekci přetížení vyřazuje úlohy v pořadí rostoucích hodnot důležitosti, dokud přetížení nepomine, a pravidlo C obnovuje v pořadí klesajících hodnot důležitosti vyřazené úlohy, je-li na procesoru detekován dostatek času k jejich včasnému provedení.

Závěr

V následujícím dílu seriálu bude diskutována možnost řešit nedostatek výpočetního výkonu použitím většího počtu procesorů. Budou představeny základní problémy v této oblasti a jejich možná řešení.

Poděkování

Tento článek byl vypracován v rámci projektu Centrum excelence IT4Innovations, reg. č. CZ.1.05/1.1.00/02.0070, podporovaného Operačním programem Výzkum a vývoj pro inovace, financovaného ze strukturálních fondů EU a ze státního rozpočtu ČR, projektu Výzkum informačních technologií z hlediska bezpečnosti (CEZ MSM 0021630528) a grantu BUT FIT-S-11-1.

Literatura:

- [1] BUTAZZO, G. C.: *Hard Real-Time Computing Systems, Predictable Scheduling Algorithms And Applications*. Kluwer, 1997, pp. 229–249, ISBN 0-7923-9994-3.
- [2] COTTET, F. – DELACROIX, J. – KAISER, C. – MAMMERI, Z.: *Scheduling in Real-Time Systems*. John Wiley & Sons, 2002, ISBN 0-470-84766-2.
- [3] ČIŽINSKÝ, V.: *Implementace pokročilých mechanismů plánování množin RT úloh běžících pod μ COS-II*. Diplomová práce, FIT VUT v Brně, 2010.
- [4] STRNADEL, J.: *Návrh časově kritických systémů I: specifikace a verifikace*. Automa, 2010, roč. 16, č. 10, s. 42–44, ISSN 1210-9592.
- [5] STRNADEL, J.: *Návrh časově kritických systémů II: úlohy reálného času*. Automa, 2010, roč. 16, č. 12, s. 18–19, ISSN 1210-9592.
- [6] STRNADEL, J.: *Návrh časově kritických systémů III: prioritá úloh*. Automa, 2011, roč. 17, č. 2, s. 50–52, ISSN 1210-9592.
- [7] STRNADEL, J.: *Návrh časově kritických systémů IV: realizace prostředky RTOS*. Automa, 2011, roč. 17, č. 4, s. 58–60, ISSN 1210-9592.
- [8] STRNADEL, J.: *Plánování úloh v systémech RT – I: závislé úlohy*. Automa, 2012, roč. 18, č. 10, s. 42–45, ISSN 1210-9592.
- [9] STRNADEL, J.: *Plánování úloh v systémech RT – II: neperiodické úlohy*. Automa, 2012, roč. 18, č. 11, s. 44–46, ISSN 1210-9592

Ing. Josef Strnadel, Ph.D.,
Centrum excelence IT4Innovations,
Fakulta informačních technologií,
Vysoké učení technické v Brně
(strnadel@fit.vutbr.cz)

ADRESÁŘ VYDAVATELSTVÍ

Ředitel: Ing. Emil Širůček

Adresa: Pod Vodárenskou věží 4, 182 08 Praha 8
tel.: 286 583 011-12, 266 052 804, fax: 284 683 022
e-mail: automa@fccgroup.cz, www.automa.cz

Šéfredaktor: Ing. Petr Bartošik

Zástupce šéfredaktora: Ing. Eva Vaculíková

Redakce: Petr Špůr, Milena Kočišová

Speciální projekty: Ing. Karel Suchý, Ing. Ladislav Šmejkal, CSc.

Oborná spolupráce:

Ing. Karel Bílek, Bernecker + Rainer Industrie Elektronik, Ges. m. b. H.,
Ing. Miroslav Dub, CSc., Sidat, spol. s r. o., **prof. RNDr. Ing. Petr Fiala, CSc., MBA**, katedra ekonomie Vysoké školy ekonomické v Praze, **Ing. Otto Havle, CSc., MBA**, FCC průmyslové systémy, s. r. o.,
doc. Ing. Petr Horáček, CSc., FEL ČVUT v Praze, **Ing. Zdeňek Hurák, Ph.D.**, katedra řídicí techniky FEL ČVUT v Praze, **doc. Ing. Josef Janeček, CSc.**, katedra řídicí techniky Fakulty mechatroniky a mezioborových studií TU Liberec, **doc. Ing. Karel Kadlec, CSc.**, ústav fyziky a měřicí techniky, Fakulta chemicko-inženýrská, Vysoká škola chemicko-technologická v Praze, **Ing. Petr Kašík, Amit, spol. s r. o.**, **doc. Dr. Ing. Vladimír Kebo**, institut ekonomiky a systémů řízení, Hornicko-geologická fakulta, VŠB-TUO Ostrava, **Ing. Marie Martinásková, Ph.D.**, ústav přístrojové techniky, Fakulta strojní ČVUT v Praze, **prof. Ing. Vladimír Mařík, DrSc.**, katedra kybernetiky FEL ČVUT v Praze, **doc. Ing. Pavel Nahodil, CSc.**, katedra kybernetiky FEL ČVUT v Praze, **prof. Ing. Miloš Schlegel, CSc.**, katedra kybernetiky FAV ZČU v Plzni, **prof. Ing. Vilém Srovnal, CSc.**, katedra kybernetiky a biomedicínského inženýrství, Fakulta elektrotechniky a informatiky VŠB-TUO Ostrava, **prof. Ing. Bohumil Šulc, CSc.**, ústav přístrojové techniky, Fakulta strojní ČVUT v Praze, **prof. Ing. Vladimír Vašek, CSc.**, Fakulta aplikované informatiky, Univerzita Tomáše Bati ve Zlíně, **prof. Ing. Petr Vavřín, DrSc.**, ústav automatizace a měření FEKT VUT Brno, **prof. Ing. František Zezulka, CSc.**, ústav automatizace a měřicí techniky FEKT VUT Brno

Inzerce: Ladislava Hošmáňková

Administrace časopisu: Ing. Petra Huňová

Sazba a grafická úprava: Dana Pecháčková, Tomáš Petr

SEZNAM INZERENTŮ

Amtel spol. s r. o.	29
B+R automatizace, spol. s r. o.	4, 36
Cognex Germany	3, 4
Distrelec GmbH	obálka 1
Eaton Elektrotechnika, s. r. o.	60
Elcom a. s., divize Virtuální instrumentace	4
Euchner electric s. r. o.	20, 21, obálka 3
Infrasensor s. r. o.	21
Invensys Systems spol. s r. o.	obálka 4
NuernbergMesse GmbH	57
Omnron electronics spol. s r. o.	20
Phoenix Contact, s. r. o.	60
Profibus CZ	59
Sensorika s. r. o.	39
Schmachtl CZ spol. s r. o.	20
Schneider Electric CZ, s. r. o.	21
Siemens s. r. o.	4, 21, 22, obálka 2
Systemtronic, s. r. o.	22
Terinvest, spol. s r. o.	61
Topinfo, s. r. o.	43
Turck s. r. o.	27
Yaskawa Czech s. r. o.	55

Automa. Vydává firma FCC Public s. r. o. Přetisk je dovolen jen se svolením redakce a s uvedením pramene. Za případné závazky ke třetím stranám ručí autor. Názory autorů nemusejí být shodné se stanoviskem redakce. Vydavatel nezodpovídá za pravdivost údajů uvedených v inzerci a PR příspěvcích. Pro předplatitele v České republice provádí distribuci v zastoupení vydavatele společnost Send Předplatné, Ve Zlíbku 1800/77, 193 00 Praha 9 Horní Počernice; příjem objednávek a reklamace: tel.: 225 985 225, fax: 225 341 425, send@send.cz, www.send.cz. Pro Slovenskou republiku: Magnet Press Slovakia s. r. o., Teslova 12, P. O. Box 169, 830 00 Bratislava, tel.: +421 244 454 559 (předplatné), +421 244 454 628 (sekretariát), předplatné@press.sk, Elez, Zlatovská 27, 911 05 Trenčín, tel.: +421 326 527 672, fax: +421 327 436 536, elez@elez.sk. Mediaprint-Kapa Pressegrasso, a. s., odd. inej formy predaja, Vajnorská 137, P. O. Box 183, 830 00 Bratislava 3, tel.: +421 244 458 821, +421 244 458 816, +421 244 442 773, fax: +421 244 458 819, e-mail: předplatne@abompka-pa.sk a Slovenská pošta, SPT, Nám. slobody 27, 810 05 Bratislava, objednávky prijíma každá pošta a poštový doručovateľ. Objednávky do zahraničí vyřizuje Mediaservis s. r. o., Paceřická 1, 193 00 Praha – Horní Počernice, tel.: 271 199 250, kauerova@mediaservis.cz. Veškeré objednávky přijímá také redakce, která zprostředkuje i případné reklamace. Vychází 12x ročně. Tiskne Kavka Print, a. s., Ke Zdišsku 620, 250 67 Klecany, tel.: 317 070 745. Do tisku předáno 10. 12. 2012, vyšlo 14. 12. 2012. Cena časopisu: 52 Kč (dvojičko 104 Kč).

FROM CONTENTS

New products

Reliable Industrial Ethernet networks with high availability	4
Intelligent I/O module of X20 series increases machine availability	4
New line cameras from Basler Racer	4
DataMan bar code reader with widened field of view	4
New Eaton ATS-C transfer switch system	60
Programmable relay for all safety functions	60

Business

ZAT completed modernisation of further power plant in Cuba	5
Opening of Yaskawa company headquarters in Allershausen near Munich	5
Weidmüller is successful in Central and Eastern Europe	6
Moderate increase on the automation market	7

Cyber security

Opinion pool on securing of control, information, and communication systems in industrial practice	8
Duqu – hidden threat	10
Integration of security systems in buildings	13
System for assets as well as persons' position and movement tracking	16
Access system with fingerprint readers	19

Fairs and conferences

Security of industrial communication networks was important subject at it-sa 2012 fair	9
Embedded World – focused on security	12
B&R's seminar on finding of ways to efficient production	37
Conference on information systems security	54
Profession societies and associations are supporting Eurotrans fair	60

News

	18, 32, 49, 51
--	----------------

Market survey

Safety switches	20
-----------------	----

Machine and plant safety

Electromechanical safety switches and locks	23
Turck's products for machine and plant safety	26
Innovation in the field light safety barriers	28
Movable machine part running down measurements at machinery safety systems	29
Contactless safety switches and locks with RFID transponders	30
Sysmac platform with integrated safety control unit	33
Risks of formation of dangerous atmosphere in electric oven and measures for their elimination	34
Revolution in wind turbine safety	36

Sensors, transmitters, instrumentation

Examples of gas humidity measurement tasks (part 3)	38
Opus 10 TSE temperature and humidity data logger	40
MEMS technology makes production of advanced sensors possible	41
Industrial bar code readers for effective data logging for Kanban system	42

Control technology

Sysmac platform is enlarging	43
Task scheduling in real-time systems – III: system overload	44
Simatic S7-1200 – gate into automation world of future	48
Modernization of combined-cycle cogeneration gas power plant control system	50

Communications in industry

Analysis of frequency converter communications on Profibus	52
Next generation of industrial communication technology for PlantStruxure	56
Profibus and Profinet news at SPS/IPC/Drives 2012 fair	58

Calendar of events

	62
--	----

List of abbreviations

	63
--	----