

## PATTHELYZET OPERÁCIÓS RENDSZEREKBEN\*

(Összefoglaló)

Quittner Pál

Egyetemi Számítóközpont

A multiprogramozás bevezetése nagymértékben megnövelte a számítógépeken adott idő alatt elvégezhető munka mennyiségét, de az egymással párhuzamosan futó folyamatok (programok) koordinálása új problémákat vetett fel. Az egyik ilyen probléma a patthelyzet (deadlock) kialakulásának elkerülése.

Patthelyzet akkor következik be, amikor két vagy több folyamat vár olyan lefoglalt erőforrásokra, melyek normális üzemmód mellett soha nem lesznek számunkra elérhetők és ezáltal úgy tartják fel egymást, hogy egyikük sem tud továbblépni. Tipikus példa erre direkt kártyaolvasó és sornyomtató használatánál az, amikor az egyik program lefoglalta a kártyaolvasót és kéri a sornyomtatót, míg a másik már korábban lefoglalta a sornyomtatót és most kéri a kártyaolvasót.

Az ilyen és ehhez hasonló jellegű patthelyzet létrejötte jelentős mértékben megnöveli a feldolgozási költségeket. Az az idő, amíg az operátor észreveszi, hogy a gép áll, mindenképpen kárbavész. Ehhez még hozzájárulnak a patthelyzet megszüntetésének, általában valamelyik program újraindításának költségei.

Az operációs rendszerek erőforrás allokálását grafikusán általában egy erőforrás, vagy egy erőforrás–folyamat irányított gráffal szokták ábrázolni (1. ábra) [1,2]. Könnyen belátható, hogy a patthelyzetnek mindkét ábrázolási módnál szükséges feltétele a gráfon belüli kör kialakulása. A számítógépes feldolgozáshoz jól algoritmizálható az erőforrásoknak, az igényeknek és a pillanatnyilag lefoglalt erőforrásoknak mátrixban való ábrázolása [3,4].

Általánosan megfogalmazva a patthelyzet létrejöttének szükséges feltétele [1], hogy a rendszerben legyen legalább két olyan erőforrás, melyeket

- (1) több folyamat igényelhet egyidejűleg, de csak egy foglalhat le kizárólagos használati joggal,
- (2) ezeket az erőforrásokat más folyamatok ideiglenesen se tudják "kölcsönkérni",
- (3) ezeket a lefoglalt erőforrásokat a lefoglaló folyamatok akkor se engedjék el, ha más erőforrások hiánya miatt várakozó helyzetbe kerültek.

A patthelyzet kivédésére szolgáló módszerek ezen szükséges feltételek valamelyikét szüntetik meg állandóan vagy ideiglenesen.

\*Az előadás részletes szövege az Automatizálás 1975 júniusi számának 18. oldalán jelent meg.

Az *eleve kizárás* módszere nem engedi meg, hogy több aktiv folyamat igényelhesse ugyanazokat az erőforrásokat. A nagy számítógépgyártó cégek leggyakrabban ezt az algoritmust alkalmazzák. Legegyszerűbb formájában az üzemező (job scheduler) addig nem indít el egy folyamatot, amíg a futás folyamán szükséges és az (1) – (3) feltételeket kielégítő összes erőforrást hozzá nem rendelte.

Az *elkerülés* módszere elvileg megengedi a folyamatok indításakor a patthelyzet létrejöttének veszélyét. Az egyes erőforrások tényleges hozzárendelésénél azonban futás közben dinamikusan megvizsgálja, hogy az így kialakuló erőforrás szétosztásánál kialakulhat-e patthelyzet, és ha igen, akkor a hozzárendelést nem engedélyezi [3,4].

Az *észlelés és megszüntetés* módszerénél a rendszer megengedi a patthelyzet kialakulását, de állandóan figyeli, hogy az bekövetkezett-e. Amennyiben igen, akkor lefoglalt erőforrások felszabadításával oldja ezt fel.

A modern számítógépeknél a hardware erőforrások legnagyobb részére az (1) vagy (2) feltétel nem teljesül. Pattveszélyt leginkább a mágnesszalag egységek, mágnesszalag file-ok és a több felhasználó által módosítható mágneslemez file-ok jelentenek. Az 1. táblázatban összefoglaltuk, hogy a legfontosabb erőforrásoknál a patthelyzet szükséges feltételei közül melyik nem teljesül, illetve ha ezek mindegyike fennáll, akkor a három fő módszer közül melyikkel szüntetik meg a patthelyzet veszélyét. A 2. táblázatban összehasonlítottuk ezen módszerek legfontosabb előnyeit és hátrányait.

Adatbázis kezelő rendszereknél, ahol elvileg több program is módosíthatja ugyanazon adatokat új, a standard operációs rendszerek erőforrás-allokálásához képest lényegesen nagyobb nehézségek jelentkeznek. Ezek közül a legfontosabbak a következők [5]:

- Az erőforrások nincsenek egyértelműen definiálva (többszörös nevek).
- Az erőforrások dinamikusan változnak és ezért az igényeket rendszerint nem lehet előre teljes mértékben meghatározni.
- Az erőforrások (az önállóan "lezárható" rekordok) száma több nagyságrenddel nagyobb.

Ezen nehézségek miatt adatbázis kezelő rendszereknél még nem alakult ki egységes módszer a pattveszély kiküszöbölésére. Az eddig alkalmazott algoritmusok [5,6] csak ad hoc jellegű megoldások.

#### Irodalom

- [1] E.G. Coffman Jr.: Deadlocks in Computer Systems, in Operating Systems, Infotech State of the Arts Report. 1972, 353.o.
- [2] A.J.T. Colin: Introduction to Operating Systems. 14. fejezet, MacDonald (American Elsevier) London, New York, 1971.
- [3] A.N. Haberman: Prevention of System Deadlocks. CACM 12 (1969) 373, 385.



- [4] R.C. Holt: Comments on Prevention of System Deadlocks. CACM 14 (1971) 36.
- [5] D.C. Chamberlain, R.F. Boyce, I.L. Traiger: A Deadlock Free Scheme for Resource Locking in a Data Base. Proc. IFIP 74, 340. o, North Holland, 1974.
- [6] A. Shoshani, A.J. Bernstein: Synchronization in a Parallel-Accessed Data Base. CACM 12 CACM 12 (1969) 604.

### Summary

Deadlock in operating systems

P. Quittner

The necessary conditions for deadlock are summarized and methods to avoid it (prevention preemption and detection) are compared. Special deadlock problems arising from simultaneous updating in a data base are discussed.

### Резюме

Наличие "пата" в операционных системах

Пал Квиттнер

В статье обобщаются условия необходимые при "пате" и сравниваются различные методы его обхода (а именно исключение, обход, изъятие). В статье также обсуждаются специальные проблемы, возникающие при параллельной модификации базисов данных.

ERŐFORRÁSOK TULAJDONSÁGAI A PATTHELYZET SZEMPONTJÁBÓL

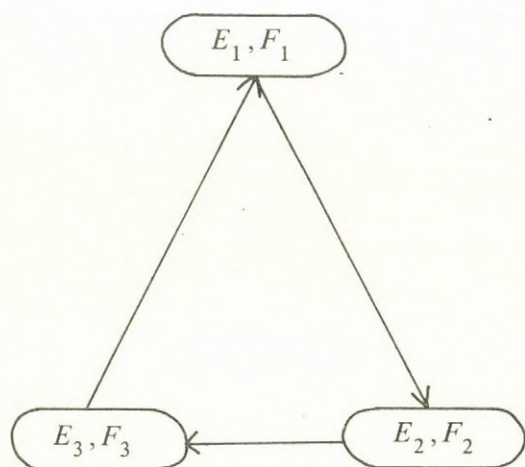
Erőforrás	Patthelyzet szükséges feltételei közül melyik nem teljesül	Patthelyzet megakadályozásának módszere
Központi memória	Kölcsönkérhető, illetve elengedi (csak roll-in roll-out vagy vagy virtuális tárnál)	Eleve kizárás
Processzor (CPU)	Kölcsönkérhető a megszakítási rendszer révén	
Csatornák	Elengedi	
Lassu perifériák	Közös virtuális I/O egységek nem foglalhatók le kizárólagos használatra	Megszüntetés (az operátor elveszi)
Mágnesszalag egységek és file-ok	A file megnyitása előtt és lezárása után kölcsönkérhetők, illetve elengedheti a rendszer	Eleve kizárás, elkerülés
Mágneslemez egységek	Közösen használhatók	
Mágneslemez file-ok	Olvasáshoz közösen használhatók	Módosításnál: eleve kizárás vagy elkerülés
Rendszerprogramok és táblázatok	Reentrant programok közösen használhatják	Eleve kizárás

1. Táblázat

PATTHELYZET MEGAKADÁLYOZÁSÁRA SZOLGÁLÓ MÓDSZEREK ÖSSZEHASONLÍTÁSA

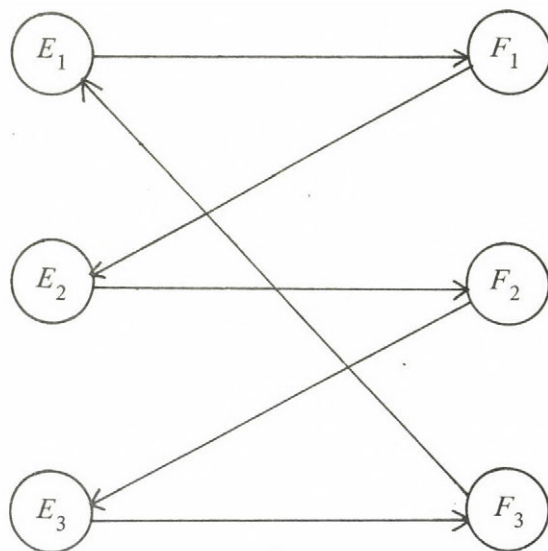
Módszer	Előny	Hátrány
Eleve kizárás	Gyors algoritmus, Soha sem lehet patthelyzet, Eredeti job-prioritás megmarad	Fölöslegesen lefoglal erőforrásokat, ezáltal a multiprogramozás lehetőségeit erősen csökkenti. Előzetes információk kellene az erőforrás igényekről.
Elkerülés	Jobban gazdálkodik az erőforrásokkal, Soha nem lehet patthelyzet	Rendszeresen megismétlendő algoritmus minden lefoglalásnál. Bonyolultabb algoritmus (különben új problémák jelentkeznek). Multiprogramozás lehetőségeit csökkenti, mert feleslegesen nem enged lefoglalni pattveszélyt előidéző erőforrásokat. Előzetes információk kellene az erőforrás-igényekről.
Észlelés	Egyszerű algoritmus, Összes szabad erőforrás allokálható, Nem kell előzetes információ az erőforrás-igényekről.	Patthelyzet kialakulhat, megszüntetése igen költséges lehet.

2. Tábázat



Folyamat	Erőforrás	
	Kért	Lefoglalt
$F_1$	$E_2$	$E_1$
$F_2$	$E_3$	$E_2$
$F_3$	$E_1$	$E_3$

a.)



b.)

1. ábra

Erőforrás allokálás grafikus ábrázolása

a.) erőforrás gráf

b.) erőforrás – folyamat gráf

A patthelyzet szükséges feltétele a gráfon belüli kör.