

Operációs rendszerek: Folyamatok, ütemezés

Balogh Ádám
Lőrentey Károly
Eötvös Loránd Tudományegyetem
Informatikai Kar
Algoritmusok és Alkalmazásai Tanszék

Balogh, Lőrentey: Operációs rendszerek – 2005–2006. évi félép

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezései
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

2006. február 27-március 6.

Folyamatok, ütemezés

2. oldal

A folyamat fogalma

- Frank Soltis: „Az idő az anyatermészet eszköze annak meggátlására, hogy minden egyszerre történjen. A számítógép eszköze ugyanerre a célra a folyamat.”
- A folyamat (*process*) egy futó program a környezetével együtt
- Környezet:
 - Lokális: kizárólag a folyamat foghatja fel és változtathatja meg
 - Globális: más folyamatok is felfoghatják és megváltoztathatják

Balogh, Lőrentey: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

3. oldal

Folyamat és program

- A program permanensen létezik, míg a folyamat tranziens, egy program elindulásakor jön létre, és annak befejeződésekor megszűnik
- A program elágazásokat, ciklusokat is tartalmazhat, a folyamat lineáris
- A programnak csak egy környezete van: a külvilágnak a program által felfogható és megváltoztatható része
- A folyamat környezete bővebb

Balogh, Lőrentey: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

4. oldal

Egy folyamat állapotai (1)

- Alapvetően kétféle állapot:
 - Aktívan használ bizonyos erőforrásokat
 - Várakozik egy vagy több erőforrásra
- Adott erőforrás szempontjából
 - Fut: aktívan használja
 - Futáskész: várakozik rá
 - Blokkolt: közömbös
- Ha egy folyamat valamely erőforrás szempontjából blokkolt, akkor kell lenni olyannak, amelyen futáskész

Balogh, Lőrentey: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

5. oldal

Egy folyamat állapotai (2)

- Kiemelt erőforrás: központi feldolgozó egységek
 - Fut: valamely központi egység végrehajtja az utasításait
 - Futáskész: arra vár, hogy egy központi egységre kerüljön
 - Blokkolt: valamely a központi egységtől különböző erőforrásra vár
- Blokkolt állapotból többféle lehet aszerint, hogy milyen erőforrásra vár

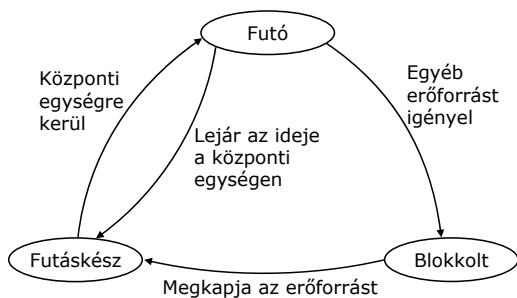
Balogh, Lőrentey: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

6. oldal

Állapotátmenetek



2006. február 27–március 6.

Folyamatok, ütemezés

7. oldal

Folyamatok megvalósítása (1)

- Processzor nem ismeri a folyamatokat:
 - Utasítássorozatot hajt végre
 - Processzor állapota változik a végrehajtás során
- Egy processzoron egyszerre egy folyamat lehet futó állapotban
 - Ha ez a folyamat leváltásra kerül, meg kell őrizni a processzor állapotát addig, amíg vissza nem kapja a processzort
 - Utasításmutató
 - Regiszterek értéke

2006. február 27–március 6.

Folyamatok, ütemezés

8. oldal

Folyamatok megvalósítása (2)

- Folyamatváltás menete:
 - Megszakítás, kivétel vagy rendszerhívás történik
 - Processzorállapot elmentésre kerül
 - Kezelőrutin:
 - Úgy dönt, hogy a folyamatot le kell váltani
 - Az elmentett állapotot más területre menti
 - Másik folyamat állapotát tölti be az előző folyamat elmentett állapota helyére
 - Visszatér
 - Processzorállapot visszatöltődik, de már a másik folyamaté

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félév

2006. február 27–március 6.

Folyamatok, ütemezés

9. oldal

Folyamatok megvalósítása (3)

- Probléma: processzor állapotához tartozik a gyorsítótárak tartalma
 - Nem hozzáférhető utasítások által ⇒ nem menthető el
 - Új folyamatnál is a régi állapotok lesznek érvényesek ⇒ újratöltődésig lecsökken a teljesítmény
- Folyamatváltás költségei:
 - Többszörös állapotmentés és visszatöltés
 - Gyorsítótárak újratöltése
- Látható: gyakori váltás ⇒ magas overhead

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félév

2006. február 27–március 6.

Folyamatok, ütemezés

10. oldal

Folyamatok megvalósítása (4)

- Többprocesszoros rendszerek:
 - Processzorok fizikailag különbözők lehetnek ⇒ nem minden folyamat fut mindegyiken
 - Processzorok feladata különböző lehet ⇒ nem minden folyamat futhat mindegyiken
 - Gyorsítótárak megőrizhetnek adatokat ⇒ ha egy folyamat újra sorra kerül, érdemes ugyanarra a processzorra tenni
 - Megvalósításban fordított szemlélet uralkodik: minden processzoron az „ismerős” (utóbbi időben futtatott) folyamatok előnyt élveznek

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félév

2006. február 27–március 6.

Folyamatok, ütemezés

11. oldal

A folyamatleíró (1)

- A folyamatleíró egy rekord, amely a folyamat adatait tartalmazza
- Lehetséges mezői:
 - Folyamatazonosító
 - Numerikus
 - Szöveges (folyamatnév)
 - Folyamattípus
 - Aktuálisan futó program adatai
 - Program neve, vagy más azonosítója
 - Tárterületek, elmentett processzoradatok
 - Hardver támogatás lehetséges (pl. Intel Pentium TSS)
 - Tulajdonos felhasználó

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félév

2006. február 27–március 6.

Folyamatok, ütemezés

12. oldal

A folyamatleíró (2)

- További lehetséges mezők:
 - Hozzáférési jogok a különböző erőforrásokhoz
 - Gyakran nem külön mező, hanem csak a tulajdonostól függ
 - Felhasználó/rendszergazda modell
 - Különböző jogosultságok
 - Néhány rendszerben a folyamat jogai eltérhetnek a tulajdonos felhasználó jogaitól
 - Korlátok, kvóták
 - Statisztikai adatok
 - Kommunikációs adatok
 - Ütemezési adatok (pl. prioritás)

2006. február 27-március 6.

Folyamatok, ütemezés

13. oldal

A folyamatleíró (3)

- A mezők értékét minden folyamat keletkezésekor meg kell határozni
 - Rendszerszinten definiált alapértelmezett érték (esetleg különböző típusú folyamatokhoz különböző értékek)
 - Felhasználónál definiált alapértelmezett érték
 - Hierarchikus folyamatok:
 - Folyamatok fastruktúrába szervezhetők
 - Szülő-gyerek kapcsolat
 - Gyerek a szülő mezőinek értékét örökli, majd ezek változnak
 - Külön mező: mutató a szülőre

2006. február 27-március 6.

Folyamatok, ütemezés

14. oldal

A folyamatábrázat

- Rendszer összes folyamatleíróját tartalmazó (konténer) adatszerkezet a folyamatábrázat
 - Vektor
 - Asszociatív adatszerkezet (pl. hasítótábla) a folyamat azonosítója szerint
- A folyamatábrázat inicializálása:
 - Rendszerbetöltő beleírja a legelső folyamatot
 - Kezdeti táblázat a háttértárról töltődik be, ami a legelső folyamatot tartalmazza

2006. február 27-március 6.

Folyamatok, ütemezés

15. oldal

Szálak

- Egy programnak az egymással párhuzamosan futó részei a szálak
- A szálak összetartoznak:
 - Programozó tud róluk, amikor a programot írja
 - Sok közös adat: például felhasználó, jogok, korlátok, statisztikai adatok stb.
 - Intenzívebb kommunikáció egymással ⇒ hatékonyabb megvalósítást igényel
- Bizonyos adatokat szálanként kell nyilvántartani: pl. processzoradatok

2006. február 27-március 6.

Folyamatok, ütemezés

16. oldal

Szálak megvalósítása

- Egyik megvalósítás: egy folyamat – több szál (*green thread*)
 - Probléma: egyik szál blokkolódik ⇒ egész folyamat blokkolódik
- Másik megvalósítás: egy folyamat – egy szál (*native thread*)
 - Problémák: költséges kommunikáció, költséges kapcsolat szálak között, erőforrások egyenlőten elosztása
- Kompromisszum: operációs rendszer megkülönbözteti a többszálú folyamatot

2006. február 27-március 6.

Folyamatok, ütemezés

17. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. **A Linux és más Unix-ok folyamatai**
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

2006. február 27-március 6.

Folyamatok, ütemezés

18. oldal

A Linux folyamata

- Megfelel az általános folyamat-fogalomnak
- Egy futó program az összes adatával

Balogh, Lőrinc: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

19. oldal

Linux folyamatok állapottai

- Futó: A Linuxban mind a futó, mind a futáskész állapot ide tartozik
- Várakozó: A blokkolt állapotok nagy része ide tartozik
 - Megszakítható: közben fogadhat szignált
 - Megszakíthatatlan: közvetlenül a hardvertől függő feltételre vár
- Megállított: szignál hatására megállt, másik szignállal folytatható
- Zombi: befejeződött, de a folyamatábrázatban maradt (halott)

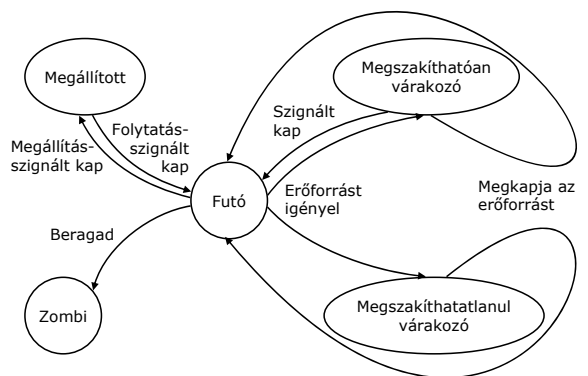
Balogh, Lőrinc: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

20. oldal

Állapotátmenetek Linux-ban



Balogh, Lőrinc: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

21. oldal

A Linux folyamatleírója (1)

- Azonosítók (numerikusak):
 - Folyamat-azonosító
 - Szál-azonosító
 - Felhasználó-azonosító
 - Felhasználócsoporthoz tartozó-azonosító
- Állapot
- Processzoradatok (elmentett regiszterek, utasításmutató, veremk stb.)
- Ütemezési információk
- Memóriakezelési információk

Balogh, Lőrinc: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

22. oldal

A Linux folyamatleírója (2)

- Állományrendszer információk
- Órák
 - Felhasználói és rendszer védelmi szinten eltöltött, valamint összes idő
 - Ébresztőórák
- Folyamatközi kommunikációs adatok
- Mutatók
 - Szülőre
 - Testvérekre
 - Gyerekekre

Balogh, Lőrinc: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

23. oldal

Folyamatok keletkezése

- Nincs folyamatot létrehozó rendszerhívás
- Betöltődéskor *init* folyamat, azonosítója 1
- *fork* rendszerhívás: másolatot készít a folyamatból, gyerekfolyamatként
- Programok indítása (folyamat régi programja helyett): *execve*
- Speciális célú program: shell (héj)
 - Feladata más programok indítása, valamint lehetnek belső parancsai is
 - Programindítás: *fork*, majd másolatban *execve*

Balogh, Lőrinc: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

24. oldal

Szálak Linux-ban

- clone rendszerhívás
 - fork-hoz hasonló
 - Új folyamat megoszthatja a régivel a
 - Memóriaterületét
 - Állományrendszer-információit
 - Állományleíróit
 - Szignálkezelőit
 - Azt, hogy nyomkövetik-e
 - Folyamatazonosítóját (szálozonosító egyedi)
 - Szülő folyamatát
 - Folyamatok szálcsoportokba szervezhetők
- POSIX: `pthread_create`

2006. február 27–március 6.

Folyamatok, ütemezés

25. oldal

Más Unix rendszerek

- Folyamat fogalma minden Unix-ban nagyon hasonló
- Állapotaik, állapotátmeneteik és keletkezésük is hasonló
- Állományleírók különbözhetnek
- Szálak megvalósítása különbözhet

2006. február 27–március 6.

Folyamatok, ütemezés

26. oldal

Szálak Solaris-ban (1)

- Könnyűsúlyú folyamatok: *kernel* szinten ütemeződnek, környezetük felhasználói szinten változhat
- Szálakhoz külön-külön tartozhatnak könnyűsúlyú folyamatok
- Ha egy blokkolódik: nem probléma
- Belső kommunikáció: felhasználói szinten
 - Emiatt blokkolódik: felhasználói szinten lecserélhető a szál
 - Hatékonyabb, mint az előző megoldások

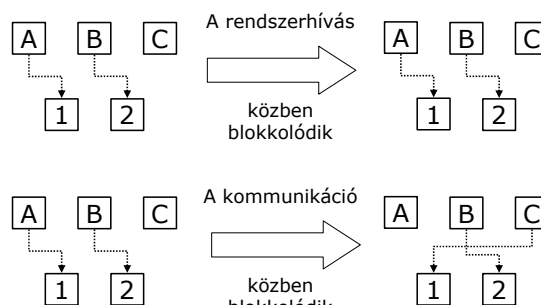
Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

27. oldal

Szálak Solaris-ban (2)



Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

28. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. **Az OpenVMS folyamatai**
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

29. oldal

Az OpenVMS folyamata

- Folyamat és a futó program fogalma különbözik
- Folyamat
 - Egyszerre két program lehet benne aktív
 - Felhasználói program (*user* védelmi szinten)
 - Shell program (*supervisor* védelmi szinten)
 - Felhasználói program dinamikusan cserélődhet
- Futó program neve: *image*
- Ritkábbak a folyamatok létrehozásának és megszüntetésének műveletei (mivel költségesebbek)

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

30. oldal

A folyamatok állapotai (1)

- Futó: *CUR*
- Futáskész: *COM*
- Alvó: *HIB*
 - *\$HIBER* rendszerhívással blokkolta magát
- Felfüggesztett: *SUSP*
 - *\$SUSPND* rendszerhívással más blokkolta
- Lokális eseményjelzőre várakozó: *LEF*
 - I/O műveletek
- Közös eseményjelzőre várakozó: *CEF*
 - Folyamatközi kommunikáció

2006. február 27-március 6.

Folyamatok, ütemezés

31. oldal

A folyamatok állapotai (2)

- Különböző laphibák (ld. később) miatt várakozó: *PFW, FPG, COLPG*
- Processzorok közötti kölcsönös kizárás vagy erőforráshiány miatt várakozó: *MWAIT*
 - Tovább osztható sok különböző alállapotra: *MUTEX* és *RW** állapotok

2006. február 27-március 6.

Folyamatok, ütemezés

32. oldal

A folyamatok hierarchiája

- Unix-szal ellentétben a folyamatok nem alkotnak fastruktúrát
- Kétféle folyamat:
 - Leválasztott folyamat: önálló, nincs szülője
 - Alfolyamat: valamely más folyamat a szülője
- Szülő megszűnések az alfolyamat is megszűnik
- Egy leválasztott folyamat az összes alfolyamatával együtt: munka (*job*)

2006. február 27-március 6.

Folyamatok, ütemezés

33. oldal

A folyamatleíró (1)

- Három részből áll:
 - Fix hosszú rész: folyamatvezérlő blokk: *PCB*
 - Változó hosszú rész: folyamat-fejléc: *PHD*
 - Munka adatai: munka információs blokk: *JIB*
- A *PCB*-ben van egy mutató a másik kettőre
- A *PCB* további mezői:
 - Folyamatazonosító: numerikus
 - Folyamatnév: szöveges
 - Állapot

2006. február 27-március 6.

Folyamatok, ütemezés

34. oldal

A folyamatleíró (2)

- A *PCB* további mezői (folytatás):
 - Kommunikációs, szinkronizációs és I/O vezérlő adatok (eseményjelzők, *AST-k*)
 - Kvóták és korlátok
 - Privilegiumok
 - Folyamat jogai különbözhetnek a tulajdonos felhasználó jogaitól
 - Prioritás (ld. Később)
- A *PHD* ezenkívül a processzoradatokat, adatokat az ütemezéshez és memória-kezeleléshez, valamint statisztikákat

2006. február 27-március 6.

Folyamatok, ütemezés

35. oldal

A folyamatleíró (3)

- A *JIB* tartalma:
 - Felhasználónév
 - Egész munkára közösen vonatkozó korlátok és kvóták
- A munka fogalma és a *JIB* segítségével a korlátozások és kvóták árnyaltabban állíthatók be:
 - Felhasználó szintű korlátok és kvóták
 - Munka szintű korlátok és kvóták
 - Folyamat szintű korlátok és kvóták

2006. február 27-március 6.

Folyamatok, ütemezés

36. oldal

A folyamattáblázat

- PCB-k (mivel fix hosszúak): rögzített hosszú vektorban
- Index: A folyamatazonosító néhány alsó bitje
- Üres helyek: *NULL* folyamatok
- Ezen kívül a nem *NULL* folyamatok között két irányú láncolt lista

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

37. oldal

Folyamatok keletkezése

- Kezdeti tábla a háttértárról töltődik be, benne néhány folyamattal
- *JOB_CONTROL*: ha olyan terminálról érkezik adat, ami nem tartozik folyamathoz, elindít egy folyamatot *LOGINOUT* programmal
- *\$CREPRC* rendszerhívással bármely folyamat létrehozhat újabb al- vagy leválasztott folyamatot

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

38. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. **A Windows folyamatai**
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

39. oldal

A Windows folyamata

- Folyamat fogalma leginkább a Unix-nál megismertre hasonlít
- Folyamat helyett a szál fogalma jelentős: folyamat szerepe elsősorban az, hogy a szálak tőle öröklik a paramétereiket
- Munka: folyamatok csoportja, hasonló adatokat tartalmaz, mint OpenVMS-nél

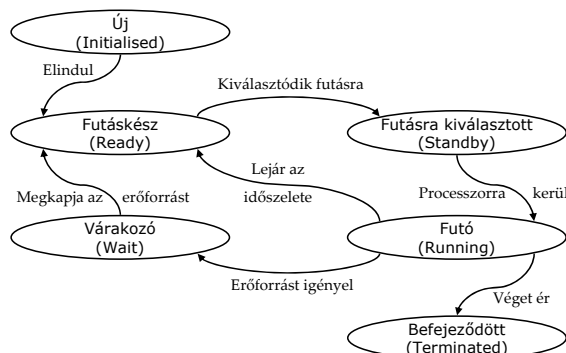
Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

40. oldal

Folyamatállapotok Windows-ban



Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

41. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. **Az OS/400 folyamatai**
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27-március 6.

Folyamatok, ütemezés

42. oldal

Az OS/400 folyamatai

- Három szint:
 - Taszk (SLIC objektum): egy szál, amit ütemezni lehet
 - Folyamat (MI objektum): tartalmaz egy szálat, és a hozzájuk tartozó memóriaterületeket
 - Munka (OS/400): hagyományos értelemben vett folyamatnak felel meg, egy folyamatot tartalmaz, ami cserélődhet

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

43. oldal

A taszk leírója

- Prioritás (ütemezéshez)
- Időszel (ütemezéshez)
- Bittérképek többprocesszoros működéshez: melyiken fut (aktivitás), melyeken futhat (engedélyezés) és ezek közül melyeken futott az utóbbi időben (affinitás)

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

44. oldal

A folyamat leírója

- Mutató a taszkra
- Táradatak
 - Processzoradatok
 - Memóriaadatok
 - Állományrendszer-adatok
- Üzenetsorok (folyamatközi kommunikációhoz)

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

45. oldal

A munka leírója

- Munkaazonosító (numerikus)
- Munkanév (szöveges)
- Felhasználónév
- Mutató a folyamatra
- Munkaleíró (JOBID OS/400 objektum): a munka nem futásidejű paraméterei
- Korlátok
- Alrendszer neve

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

46. oldal

Az alrendszer fogalma

- Az alrendszer egy előre definiált környezet, melynek célja:
 - Erőforrások felosztása munkák csoportjai között
 - Tárterületek
 - Munka-belépési eszközök
 - Alapértelmezett értékek meghatározása azonos csoportba tartozó munkák számára
 - Irányítási táblázat alapján további alcsoportok, különböző alapértelmezett értékekkel, és tovább osztott erőforrásokkal
- Az alrendszer aktív, nemcsak leírás

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

47. oldal

A munkatáblázat

- Rögzített hosszú vektor
- Dinamikusan növekedhet a mérete
 - Növelés költséges művelet!
- Alrendszerek is hasonló táblázatban vannak eltárolva

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

48. oldal

Munkák indulása

- Alrendszer:
 - Belépési pontok: terminálok, munkasorok stb.
 - Alrendszer indítja a munkákat, melyek a belépési pontokon keresztül érkeznek be
 - Meghatározott szabályok és korlátok az egyes belépési pontokra
- Munkák:
 - Különböző típusok: interaktív, köteget stb.
 - Programjaik elhelyezhetik elindítandó munkák leírását munkasorokban

2006. február 27–március 6.

Folyamatok, ütemezés

49. oldal

Szálak

- Eredeti megvalósítás: külön munkák közös memóriaterülettel
 - Előre gyártott üres munkák gyorsítják a szálak létrehozását
- Későbbi megvalósítás: egy folyamathoz több taszk is tartozhat

2006. február 27–március 6.

Folyamatok, ütemezés

50. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. **Az ütemezés fogalma és céljai**
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

2006. február 27–március 6.

Folyamatok, ütemezés

51. oldal

Az ütemezés fogalma

- Ütemezés: a végrehajtó egységek, mint kiemelt erőforrások elosztása az azt igénylő folyamatok között
- Általános eset: folyamatok száma legalább egy nagyságrenddel nagyobb, mint a végrehajtó egységeké
- Szuperszámítógépek: Gyakran egy folyamat szálait kell ütemezni, annyit, ahány végrehajtó egység van: feladat inkább a szálak elhelyezése úgy, hogy a kommunikáció hatékony legyen

2006. február 27–március 6.

Folyamatok, ütemezés

52. oldal

Az optimum fogalma

- Sok különböző feladat \Rightarrow sok különböző szempont
- Hasonló a különböző tervezési célokhoz
 - Gyakran ellentmondások
- Gyakran a felhasználási kategóriákkal is összefüggésben van
 - Egy kategórián belül is többféle szempont lehetséges
- Továbbiakban: egy végrehajtó egységet feltételezünk

2006. február 27–március 6.

Folyamatok, ütemezés

53. oldal

Néhány jelölés

- i . folyamat érkezési ideje: b_i
- i . folyamat befejeződése: e_i
- i . folyamat feldolgozási ideje: p_i
- i . folyamat határideje: d_i

2006. február 27–március 6.

Folyamatok, ütemezés

54. oldal

Ütemezési célok (1)

- Igazságosság: minden folyamat számára azonosak a feltételek
 - Mindennapi életben példa: sorban állás

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi file

2006. február 27-március 6.

Folyamatok, ütemezés

55. oldal

Ütemezési célok (2)

- Átlagos átfutási idő minimalizálása: a folyamatok átlagban minél előbb jussanak el indulásuktól a befejeződésükig
 - Legalább akkora, mint az összes munka elvégzéséhez szükséges idő
 - Munkák száma adott \Rightarrow átlag helyett az összeg is használható
 - Képletben: $\min \sum(e_i - b_i)$

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi file

2006. február 27-március 6.

Folyamatok, ütemezés

56. oldal

Ütemezési célok (3)

- Válaszidő minimalizálása: hasonló az átfutási időhöz, de más környezetben
 - Rövid parancs (pl. könyvtárlista): gyors válasz
 - Hosszú parancs (pl. fordítás): válasz többet késhez
 - Az átfutási idő és az elvégzéshez szükséges idő arányának minimalizálása
 - Képletben: $\min \sum[(e_i - b_i)/p_i]$

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi file

2006. február 27-március 6.

Folyamatok, ütemezés

57. oldal

Ütemezési célok (4)

- Átlagos késés minimalizálása: átlagosan minél kevesebbet csúszzanak a folyamatok egy megadott határidőn túl
 - Negatív csúszás nincs, csak nulla!
 - Képletben: $\min \sum \max\{(e_i - d_i), 0\}$

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi file

2006. február 27-március 6.

Folyamatok, ütemezés

58. oldal

Ütemezési célok (5)

- Legnagyobb késés minimalizálása: ugyanaz, mint az előző, csak átlag helyett a maximumot minimalizáljuk
- Elkészülés határidőre: az előző szigorítása: ha annál nem nulla az optimum, nincs megoldás

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi file

2006. február 27-március 6.

Folyamatok, ütemezés

59. oldal

Ütemezési célok (6)

- Különböző fontosságú folyamatok:
 - Súlyozás: egyes folyamatok nagyobb súlyal számítanak valamely optimum kiszámításánál
 - Például legrövidebb átfutási idő: $\min \sum w_i(e_i - b_i)$
 - Prioritások:
 - Mindig csak az aktuálisan legnagyobb prioritásúak optimumára törekszünk
 - Ha a legnagyobb prioritásúakat már optimálisan ütemeztük, akkor a következő legnagyobb prioritásúakkal folytatjuk ugyanígy

Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi file

2006. február 27-március 6.

Folyamatok, ütemezés

60. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. **Folyamatok típusai és ütemezéseik**
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

2006. február 27-március 6.

Folyamatok, ütemezés

61. oldal

Kötegelt folyamatok

- Felhasználótól menet közben nem vár adatot
- Futásukhoz szükséges idő: néhány másodperc és több nap között
- Fontos: elkészülés határidőre, vagy legalább minimális átlagos vagy legnagyobb késés, esetleg átlagos átfutási idő
- Nem fontos: válaszidő
- Lehetnek prioritások vagy súlyok
- Példák: pénzügyi intézetek éjszakai vagy hétvégi feldolgozásai (pl. kamatok könyvelése), mentések, hálózati letöltések, víruskeresés

2006. február 27-március 6.

Folyamatok, ütemezés

62. oldal

Interaktív folyamatok

- Futás közben állandó kommunikáció a felhasználóval
- Tranzakció: két interakció (felhasználótól történő adatfogadás) közötti futás
- Fontos: válaszidő
 - Pszichológiai teszt: 0,3 s válaszidő alatt az ember „azonnali” választ érzékel
- Lehetnek különböző súlyok és prioritások
- Példák: munka a számítógép előtt, azaz szövegszerkesztés, rajzolás, bizonyos játékok, de: programok fordítása már inkább kötegelt!

2006. február 27-március 6.

Folyamatok, ütemezés

63. oldal

Valós idejű folyamatok

- Eseményekre megadott határidőn belül kell reagálni
- Lágú: csúszások megengedettek
- Szigorú: tilos csúszni
- Fontos: elkészülés határidőre, átlagos vagy legnagyobb késés minimalizálása vagy válaszidő
- Különböző fontosságú események lehetnek
- Példák: beágyazott rendszerek (szabályozás), multimédia (játékok is)

2006. február 27-március 6.

Folyamatok, ütemezés

64. oldal

Készre futtatás és megszakíthatóság

- Ha egy folyamatot addig engedünk a processzoron futni, amíg véget ér, vagy blokkolódik: készre futtatjuk
- Folyamattól menet közben is el lehet venni a processzor, azaz meg lehet szakítani
 - Gyakoribb folyamatváltás miatt így nagyobb lesz az overhead
 - Bizonyos speciális célrendszerekben architektúrális okokból nem lehetséges (időzítő hiánya, folyamat állapota nem tárolható stb.)
 - Bizonyos rendszerfolyamatoknál nem célszerű, mert a rendszer működésképtelen, amíg a folyamat be nem fejeződik

2006. február 27-március 6.

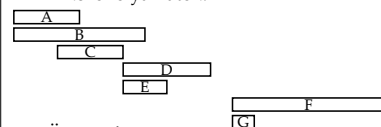
Folyamatok, ütemezés

65. oldal

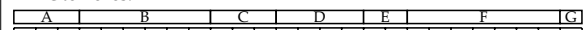
A FIFO algoritmus

- Folyamatok a rendszerbe érkezésük sorrendjében kerülnek ütemezésre
- A folyamatokat készre futtatjuk
- Példa:

Érkező folyamatok:



Ütemezés:



2006. február 27-március 6.

Folyamatok, ütemezés

66. oldal

A FIFO algoritmus tulajdonságai

- Igazságos algoritmus
- Kötegelt rendszerekben alkalmas
- Alacsony overhead
- Megvalósítás: sor adatszerkezettel
- Probléma: nagyon hosszú folyamatok az összes utánuk következőt feltartják: *kamion hatás*

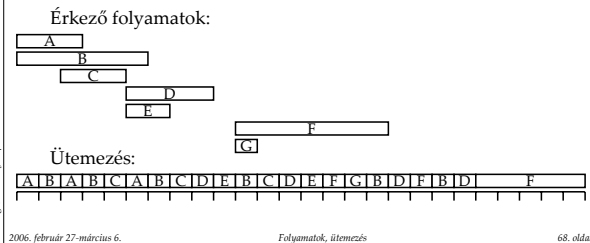
Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félév
2006. február 27–március 6.

Folyamatok, ütemezés

67. oldal

A Round-Robin algoritmus

- FIFO továbbfejlesztése: bizonyos idő (időszak, *quantum*) elteltével a folyamat végrehajtása megszakad, és a sor végére kerül
- Példa:



A Round-Robin algoritmus tulajdonságai

- Magasabb overhead (folyamatváltás)
- Interaktív folyamatokhoz is alkalmas
- Igazságos: mint a FIFO, és azonos időszak minden folyamatnak
- Megvalósítás:
 - továbbra is sor adatszerkezettel
 - programozható időzítő által generált megszakítások idézik elő a folyamatváltást

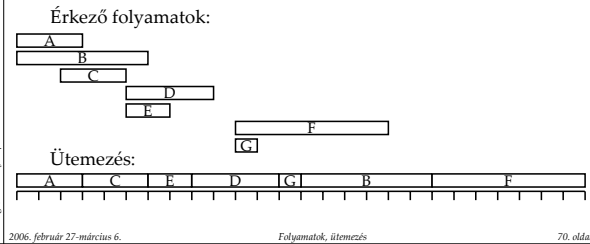
Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félév
2006. február 27–március 6.

Folyamatok, ütemezés

69. oldal

A legrövidebbet először (SJF) algoritmus

- Cél: legkisebb átlagos átfutási idő biztosítása
- Ismerünk egy felső becslést a folyamatok hosszára
- Folyamatokat méretük szerint csökkenő sorrendben, készre futtatjuk (⇒ kötegelt rendszerekben)
- Példa:



Az SJF algoritmus tulajdonságai

- Megvalósítás: prioritásos sor, ahol a sorbeli prioritás a várható futási idő reciproka
- Ha minden folyamat egyszerre indul, akkor optimális
- Ha nem egyszerre indulnak a folyamatok, akkor nem mindig biztosítja a legrövidebb átfutási időt:
 - Adott öt folyamat (A, B, C, D és E): $b_A=0, p_A=2, b_B=0, p_B=4, b_C=3, p_C=1, b_D=3, p_D=1, b_E=3, p_E=1$
 - SJF: A, B, C, D, E; átlagos átfutási idő: 4,6
 - Optimális (ha előre ismernénk, hogy melyik folyamat mikor érkezik): B, C, D, E, A; átlagos átfutási idő: 4,4
 - Ha még a várakozás is megengedett lenne: A, idle, C, D, E, B; átlagos átfutási idő: 3,6

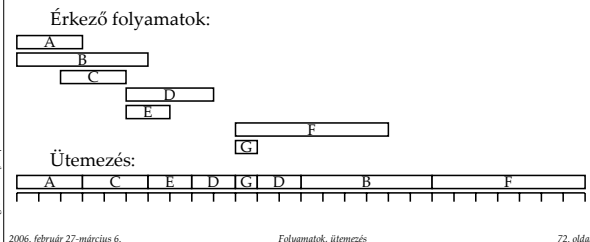
Balogh, Lőrincz: Operációs rendszerek – 2005–2006. évi félév
2006. február 27–március 6.

Folyamatok, ütemezés

71. oldal

A legrövidebb maradékot először (SRJF)

- Ha az újonnan érkezett folyamat várhatóan rövidebb, mint az éppen futtatottból hátralevő rész, akkor megszakítjuk a jelenlegit, és az újat kezdjük el futtatni: *shortest remaining job first*, SRJF
- Példa:



Az SRJF algoritmus tulajdonságai

- Valamivel magasabb overhead
- Megvalósítás: prioritásos sorba való visszatétel, vagy verem a maradékok számára
- Állítás: legrövidebb átlagos átfutási időt adja
- Példa (az előző):
 - Adott öt folyamat (A, B, C, D és E): $b_A=0$, $p_A=2$, $b_B=0$, $p_B=4$, $b_C=3$, $p_C=1$, $b_D=3$, $p_D=1$, $b_E=3$, $p_E=1$
 - SJF: A, B, C, D, E; átlagos átfutási idő: 4,6
 - Várakozással: A, idle, C, D, E, B; átlagos átfutási idő: 3,6
 - SRJF: A, B(1), C, D, E, B(3); átlagos átfutási idő: 3,4

2006. február 27–március 6.

Folyamatok, ütemezés

73. oldal

Interaktív SJF

- Átlagos válaszidőt szeretnénk minimalizálni
- Folyamatok tranzakciókból állnak \Rightarrow legrövidebbnek ígérkező tranzakciót választjuk
- Becslés: folyamatok viselkedése lassan változik \Rightarrow előző tranzakciók hosszából becslhető a következőnek a hossza
- Legutóbbi tranzakciók számítanak leginkább
- Képletben: $\tau_{n+1} = \alpha t_n + (1-\alpha)\tau_n$, ahol τ_i az i . tranzakció becslés, t_i a valós ideje, α a súlyozási faktor
- Ha $\alpha=1/2$, akkor $\tau_{n+1} = 1/2 t_n + 1/4 t_{n-1} + 1/8 t_{n-2} + \dots$
- Hibás becslés esetén a tranzakció megszaitható
- Interaktív SRJF: hasonlóan

2006. február 27–március 6.

Folyamatok, ütemezés

74. oldal

Súlyok és prioritások

- Súlyozás: SJF esetén ekkor a súlyozott átlagos átfutási időt szeretnénk minimalizálni \Rightarrow nem p_i , hanem p_i/ω_i szerint kell a folyamatokat rendezni
 - Megvalósítás: súlybeli prioritás a p_i/ω_i
- Prioritások: mindig az aktuálisan legnagyobb prioritású folyamatot ütemezzük
 - Ha több ilyen van \Rightarrow valamely előzőleg említett algoritmust alkalmazzuk, de csakis ezekre a folyamatokra
 - Megvalósítás: minden prioritáshoz külön (algoritmustól függően esetleg prioritásos) sor
 - Egyetlen prioritásos sor, ha magához az algoritmus-hoz is az kell, akkor prioritások lexikografikusan

2006. február 27–március 6.

Folyamatok, ütemezés

75. oldal

Többsoros ütemezés

- Cél: interaktív folyamatok esetén minél kisebb válaszidő, de nem tudjuk előre a tranzakciók hosszát
- Ötlet: prioritások, holtverseny esetén round robin
- Megoldás: Kezdetben minden tranzakciónak a legnagyobb a prioritása, időszelét lejártakor csökken eggyel \Rightarrow rövid tranzakciónak alacsony lesz a válaszideje
- Továbbfejlesztés: overhead csökkentése: alacsonyabb prioritásokhoz nagyobb időszelét, a legalacsonyabbhoz esetleg végtelen (FIFO)
- Tipikus megvalósítás: minden prioritáshoz külön sor

2006. február 27–március 6.

Folyamatok, ütemezés

76. oldal

Garantált ütemezés

- Fő cél: igazságosság
- Minden folyamat számára garantáljuk, hogy legalább mekkora részt kap a központi egységből
- Megvalósítás: nyomon követjük, hogy a garantált részhez képest mennyit kapott, és azt a folyamatot futtatjuk, amelyiknél ez az arány a legkisebb
- Processzorhasználat állandó nyomon követése nehéz

2006. február 27–március 6.

Folyamatok, ütemezés

77. oldal

Sorsjáték ütemezés

- A garantált ütemezés közelítése véletlenekkel
- Minden folyamat kap egy sorsjegyet
- Véletlenszerűen választunk egy sorsjegyet, és a tulajdonosát ütemezzük
- Súlyozás: folyamatok több sorsjegyet is kaphatnak
- Megfelelő véletlen generátorral „hosszú távon” igazságos (nagy számok törvénye)

2006. február 27–március 6.

Folyamatok, ütemezés

78. oldal

Valós idejű ütemezések (1)

- Események periodikusak vagy aperiodikusak
- Egyszerű eset: minden esemény periodikus
- Ütemezhetőség szükséges feltétele: ha P_i az i . esemény periódushossza: $\sum(p_i/P_i) \leq 1$
- Szigorú valós idejű rendszerek: határidő elmulasztása végzetes
 - Ekkor ez a feltétel nem elégséges
 - b_i az első előfordulás, d_i az első határidő
 - Példa: $b_1=0, b_2=1, P_1=10, P_2=10, p_1=3, p_2=3, d_1=4, d_2=5$

2006. február 27-március 6.

Folyamatok, ütemezés

79. oldal

Valós idejű ütemezések (2)

- Állandó arány:
 - i . esemény prioritása: $1/P_i$
 - Ezután prioritásos ütemezést alkalmazunk (azonos prioritások esetén FIFO-t)
- Legkorábbi határidő először:
 - Legkisebb d_i -jű folyamatot ütemezzük
- Legkisebb lazaság:
 - Legkisebb $d_i - p_i$ -jű folyamatot ütemezzük
- Mindegyik algoritmus lehet a folyamatok megszakításával vagy anélkül

2006. február 27-március 6.

Folyamatok, ütemezés

80. oldal

Kétszintű ütemezés

- Néha nem minden folyamat érhető el rövid időn belül, kevés erőforrás felhasználásával (pl. nem mind fér el a központi tárban)
- Overhead csökkentése: két ütemező
 - Egyik csak a gyorsan elérhető folyamatokat ütemezi valamelyik algoritmussal
 - Másik meghatározza, hogy mely folyamatok kerüljenek be a gyorsan elérhetőek közé

2006. február 27-március 6.

Folyamatok, ütemezés

81. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. **A Linux ütemezése**
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

2006. február 27-március 6.

Folyamatok, ütemezés

82. oldal

A Linux ütemezése (1)

- Két folyamattípus: interaktív és valós idejű
- Ütemezési stratégia: folyamatonként megadható, a valós idejűeknél: round robin vagy FIFO; interaktívknál mindig round robin
- Prioritások: mindig a legmagasabb prioritásúak közül választ (sor)
- Valós idejű: a tényleges prioritása 1000-rel nagyobb

2006. február 27-március 6.

Folyamatok, ütemezés

83. oldal

A Linux ütemezése (2)

- Időszlet: számláló, melynek kezdőértéke a prioritás
- Számláló: minden időegységben csökken eggyel, majd ha 0 lesz, akkor leváltódik
- Nincs ütemezendő folyamat: *idle* folyamat
- Többprocesszoros működés:
 - Processzor-engedélyező bittérkép
 - Magasabb tényleges prioritás azon a processzoron, ahol legutóbb futott

2006. február 27-március 6.

Folyamatok, ütemezés

84. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. **Az OpenVMS ütemezése**
10. A Windows ütemezése
11. Az OS/400 ütemezése
12. A z/VM ütemezése

2006. február 27–március 6.

Folyamatok, ütemezés

85. oldal

Az OpenVMS ütemezése (1)

- Mindhárom folyamattípust (interaktív, kötegelt, valós idejű) támogatja
- Prioritások: 32 szint, interaktív: 4-9, valós idejű 16-31, kötegelt: 3
- Minden folyamathoz külön kezdeti prioritás és aktuális prioritás
- Időszlet minden folyamatnál azonos
- Linuxhoz hasonlóan minden prioritáshoz külön sor, prioritásos round robin
- Kötegelt, interaktív: prioritás állandó

2006. február 27–március 6.

Folyamatok, ütemezés

86. oldal

Az OpenVMS ütemezése (2)

- Interaktív: ha lejár az időszlet, prioritás csökken eggyel, de nem mehet a kezdeti prioritás alá
- Várakozó állapotokhoz külön sorok
- Befejeződik a várakozás: típusától függően nő az interaktív folyamat prioritása, legfeljebb 9-ig
- Kötegelt folyamatok: felsőbb szinten vannak szabályozva: munkasorok, különböző paraméterekkel

Balogh, Lőrincz, Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

87. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. **A Windows ütemezése**
11. Az OS/400 ütemezése
12. A z/VM ütemezése

Balogh, Lőrincz, Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

88. oldal

A Windows ütemezése (1)

- Interaktív és valós idejű folyamatokat támogatja
- OpenVMS-hez hasonlóan 32 prioritási szint: valós idejű: 16-31; dinamikus: 1-15; fenntartott: 0
- Felületen csak Real-Time (24±2), High (13±2), Above Normal (10±2), Normal (8±2), Below Normal (6±2)
- Időszlet: 3-egységgel fogy minden óramegszakításnál, várakozásnál 1-gyel

Balogh, Lőrincz, Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

89. oldal

A Windows ütemezése (2)

- Időszletek különbözhetnek:
 - Előtérben lévő ablak szálainak lehet hosszabb
 - Munkák külön ütemezési osztályokban
- Minden másban (sorok, prioritás növelése és csökkentése) az OpenVMS-re hasonlít
- Különleges eset: régóta nem ütemeződött ⇨ prioritása 15-re két időszletnyi időre, majd vissza az alap-prioritásra

Balogh, Lőrincz, Operációs rendszerek – 2005–2006. évi félép

2006. február 27–március 6.

Folyamatok, ütemezés

90. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
- 11. Az OS/400 ütemezése**
12. A z/VM ütemezése

2006. február 27-március 6.

Folyamatok, ütemezés

91. oldal

Az OS/400 ütemezése (1)

- Kétféle munka: interaktív és köteget
- Egyféle taszk: két munkatípus megkülönböztetése csak felsőbb szinten történik, MI alatt csak a paraméterek mások
- Taszkok: időszület és prioritás taszkonként különbözhet
- OS/400 szint: munkák munkaleírók alapján, munkaleírókhoz osztályok, osztályokban időszület és prioritás
- Alrendszerekhez külön munkaleírók

2006. február 27-március 6.

Folyamatok, ütemezés

92. oldal

Az OS/400 ütemezése (2)

- Egytlen prioritásos sor: első taszk ütemeződik
- Ha egy folyamat túl régóta vár, vagy olyan erőforrást használ, melyre magasabb prioritású taszk vár: nő a prioritása
- Több processzor:
 - Engedélyező bittérkép
 - Affinitási bittérkép: melyiken futott utoljára
 - Processzor az első megadott számú taszkot végignézheti, ha addig nem talál megfelelőt

2006. február 27-március 6.

Folyamatok, ütemezés

93. oldal

Tartalomjegyzék

1. A folyamatokról általában
2. A Linux és más Unix-ok folyamatai
3. Az OpenVMS folyamatai
4. A Windows folyamatai
5. Az OS/400 folyamatai
6. Az ütemezés fogalma és céljai
7. Folyamatok típusai és ütemezéseik
8. A Linux ütemezése
9. Az OpenVMS ütemezése
10. A Windows ütemezése
11. Az OS/400 ütemezése
- 12. A z/VM ütemezése**

2006. február 27-március 6.

Folyamatok, ütemezés

94. oldal

A z/VM ütemezése (1)

- Nem különbözteti meg a folyamat-típusokat, rugalmasan hangolható
- A többsoros ütemezés ötletén alapul
- 3 sor, minden folyamat az elsőben indul
- Prioritás: share: az ennek megfelelő helyre kerül a sorban
- Mindig a sor elején levő ütemeződik
- Ha lejár időszülete:
 - Egész sor előrecsúszik egyvel
 - Első visszakerül az eredeti helyére (share)

2006. február 27-március 6.

Folyamatok, ütemezés

95. oldal

A z/VM ütemezése (2)

- Ha várakozni kezd: marad a sorban, csak nem ütemeződik
 - Túl sokáig várakozik (30 sec): „dormant” sorba kerül
 - „Dormant” sorban levőktől bármely erőforrás elvehető
- Ha erőforráshiány miatt kell várnia: sorához tartozó „eligible” sorba kerül
 - Erőforrások csak egyben vehetők el tőle, és egyben kell őket visszaadni

2006. február 27-március 6.

Folyamatok, ütemezés

96. oldal

A z/VM ütemezése (3)

- Ha túl sok időt tölt az első sorban, akkor átkerül a másodikba, majd a harmadikba
- Sorok között szigorúan prioritásos ütemezés
- Interakció hatására az első sorba kerül vissza
- Felső szintű ütemező: időszeleteket állítgatja az igények szerint
- Igények: share-ek: relatív és abszolút, minimum és lágú vagy szigorú maximum

Balogh, Lőrincz: Operációs rendszerek – 2006-2008. évi filip

2006. február 27-március 6.

Folyamatok, ütemezés

97. oldal

A z/VM ütemezése (4)

- További igények: tranzakciók 66-75%-a az első sorban fusson le, 0,3-0,6 sec alatt
- Ha egy folyamat túl kevés processzort kap:
nulladik sor: kárptolás
 - Nulladik sorból nem kerül át, amíg nem kárptólódik
 - Szándékosan is be lehet ragaszni folyamatokat a nulladik sorba
- Még egy különlegesség: tranzakció elején különlegesen sok processzoridőt kaphat, hogy ha rövid, gyorsan lefusson

Balogh, Lőrincz: Operációs rendszerek – 2006-2008. évi filip

2006. február 27-március 6.

Folyamatok, ütemezés

98. oldal