

Fábián Zoltán – Hálózatok elmélet

Operációs rendszerek
Multiprogramozás, hibatűrő
rendszerek

Erőforrás (Resource)

- ! Minden olyan dologi és személyi eszköz, ami egy cél eléréséhez szükséges
 - § Dologi erőforrás – Olyan eszközök, amelyek kellenek a cél eléréséhez
 - § Emberi erőforrás – Emberi munkaidő. Személyi szükséglet
- ! Megosztható erőforrás – több folyamat használja egy időben
- ! Nem megosztható erőforrás – Kizárólagosan használja egy időben az erőforrást egy folyamat – pl-. Nyomtató
- ! Kiemelt erőforrások
 - § Minden processz használja – memória, processzoridő, háttértár
- ! Preemptív erőforrás – amikor egyik folyamat elveheti a másiktól az erőforrást
- ! Non-preemptív – amikor nem elvehető az erőforrás a másiktól

Erőforráskezelés (resource management)

- ! Az erőforrásokat az erőforrás kezelő alkalmazások kezelik (resource manager)
- ! Erőforrás használatának folyamata
 - § Egy folyamat erőforrást kér az erőforráskezelőtől
 - § Az erőforrás kezelő lefoglalja az erőforrást
 - Statikus – Egyszer foglalja le az erőforrást
 - Dinamikus – Menet közben szükség szerint foglalja le az erőforrást
 - § A folyamat használja az erőforrást
 - § Ha a folyamatnak nincsen szüksége az erőforrásra, visszaadja => Az erőforráskezelő felszabadítja az erőforrást

Erőforrás kezelésre példák

- ! Windows 7 – memóriakezelés (Superfetch)
- ! Nyomtatók kezelése
- ! Firefox problémás memóriakezelése
- ! Megszakítások kezelése (IRQ)

Holtpont

- ! Olyan helyzet, amikor a processzek más processzek által lefoglalt erőforrásokra kölcsönösen várnak. Csak akkor futnak tovább a processzek, ha a szükséges erőforrások felszabadulnak.
- ! Kialakulásának feltételei
 - § Kölcsönös kizárás
 - § Foglalta várakozás
 - § Nem elvehető erőforrások
 - § Körkörös várakozás
 - § Elégséges feltétel: Ha minden erőforrás osztályban egy erőforrás van

Hotlpont kezelési stratégiák

- ! Erőforrás használati szabályokkal biztosítani, hogy holtpont ne alakuljon ki:
 - § Holtpont megelőzés (deadlock prevention)
 - § Holtpont elkerülés (deadlock avoidance)
- ! Csak a holtpont kialakulásánál avatkozunk be:
 - § Holtpont felismerés (deadlock recognition)
 - § Holtpont felszámolása (deadlock recovery)

Holtpont megelőzése 1.

! A holtpont kialakulásának valamelyik szükséges feltételének kizárása:

§ *Kölcsönös kizárást* nem lehet kiküszöbölni.

§ *Foglalva várakozás kizárása:*

Egy folyamat csak akkor kérhet új erőforrást, ha nem tart lefoglalva másikat. Alternatívák:

- Futás elején lefoglalja az összes erőforrást.
- Erőforrás-foglalás előtt a foglalt erőforrásokat felszabadítja.

Holtpont megelőzése 2.

- ! *Nem elvehető erőforrások kezelése:*
 - § erőforrások elvétele egyes folyamatoktól.
- ! *Körkörös várakozás kizárása:*
 - § erőforrások megszámozása,
 - § erőforrások csak sorrendben igényelhetők.

A holtpont elkerülése

- ! Az erőforrások óvatos allokálásával elkerülhető a holtpont kialakulása.
- ! Megvalósítás feltétele:
 - § a folyamatok erőforrás-igényéről kiegészítő információval kell rendelkezni:
 - § a folyamatok erőforrás-osztályonkénti maximális igénye

A holtpont elkerülésének módszerei

- § One Shot allocation – a futás kezdetekor minden szükséges erőforrást lefoglalunk
- § Precedencia a nem elvehető erőforrások esetén – Kategóriákba osztjuk őket => Egy kategóriából csak egyszer lehet foglalni, utána már csak magasabb szintű erőforrást lehet foglalni
- § Bankár algoritmus – amikor az alkalmazás erőforrást kér – az erőforrás-kezelő lejátsza, hogy mi történik akkor, ha megadja az erőforrást. Ha nem lesz holtpont, akkor odaadja.

Holtpont felismerése

- ! Az erőforrás-használati gráf elemzése alapján.
- ! Gráf redukciós algoritmus:
 - § Minden olyan folyamat kiválasztása, amelynek a jelenlegi igényei a szabad erőforrásokkal kielégíthetők.
 - § A folyamat által lefoglalt erőforrásokat visszaadjuk (optimista algoritmus).
 - § Újra keresünk kielégíthető folyamatot, ha nincs ilyen, de maradtak kielégítetlen folyamatok, akkor holtpont van.

Holtpont felismerése 2.

- ! Hátrány:

 - § Az algoritmus viszonylag lassú.

- ! Mikor fusson?

 - § Minden erőforrás igény teljesítésekor.

 - § Meghatározott időközönként.

A holtpont felszámolása 1.

- ! Folyamatok megszüntetésével:
 - § Minden holtpontban lévő folyamatot megszüntetünk.
 - § Csak néhány folyamatot szüntetünk meg.
 - § A megszüntetendő folyamat kiválasztásához paraméterek pl.:
 - Hány holtpont hurokban szerepel.
 - Mekkora a prioritása.
 - Mennyi ideje fut (várhatóan mennyi ideig futna még).
 - Milyen erőforrásokat tart lefoglalva.
 - Milyen erőforrásokat igényelne még

A holtpont felszámolása 2.

- ! Erőforrások elvételeivel.
 - § Probléma: melyik folyamattól és milyen erőforrásokat vegyük el.
- ! A futási eredmény megőrzéséhez szükséges ellenőrzési pontok
 - § (*checkpoint*) definiálása és
 - § a visszaállítás (*rollback*) használata.

Holtponi helyzet megoldásai

- ! Ha már kialakult a holtponi
 - § Kilőjük az egyik alkalmazást – Melyiket? Több stratégia létezik (legkevesebb fájl, legrégebben fut, legkevesebbet futott, sok erőforrást foglal, stb...)
- ! Holtponi megelőzése
 - § One Shot allocation – a futás kezdetekor minden szükséges erőforrást lefoglalunk
 - § Precedencia a nem elvehető erőforrások esetén – Kategóriákba osztjuk őket => Egy kategóriából csak egyszer lehet foglalni, utána már csak magasabb szintű erőforrást lehet foglalni
 - § Bankár algoritmus – amikor az alkalmazás erőforrást kér – az erőforrás-kezelő lejátsza, hogy mi történik akkor, ha megadja az erőforrást. Ha nem lesz holtponi, akkor odaadja.