13. előadás

Hibatűrés

3. rész

Felépülés

Bevezetés

Hibaállapotból való felépülés típusai:

- Visszatérő felépülés: korábbi helyes állapotba jutás
  (állapot rendszeres mentése, ellenőrző pontok)
- Előrehaladó felépülés: új állapotba jutás (hibajavítás)

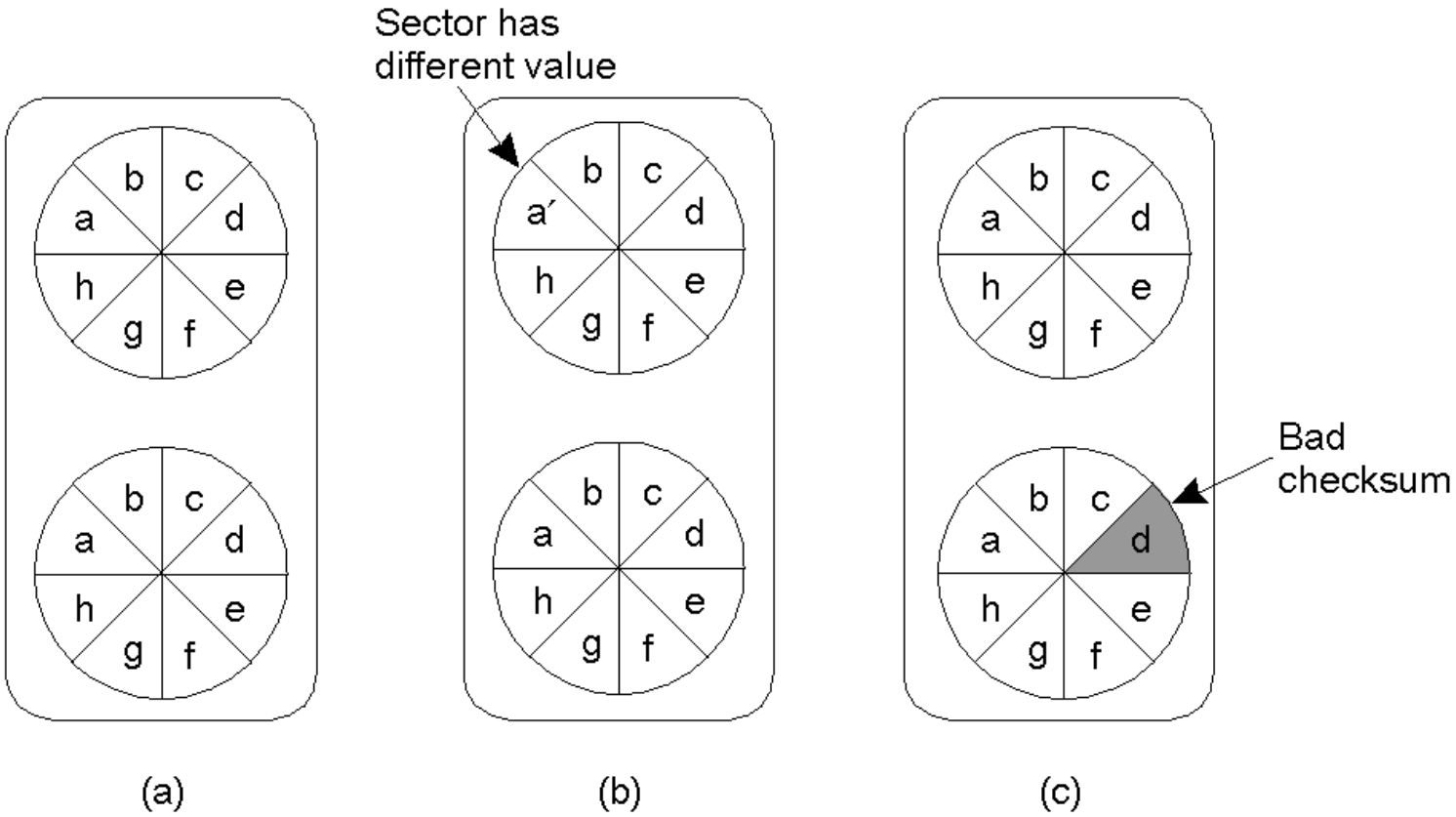
Visszatérő felépülés

- A visszaállítás költséges művelet
- Átlátszóság (felépülési ciklus)
- Nem minden művelet fordítható vissza

- Ellenőrző pontok elhelyezése
- Üzenetek naplózása
  - feladó alapú naplózás
  - címzett alapú naplózás

- Kevesebb ellenőrző pont + üzenetnaplózás

Stabil tárolás



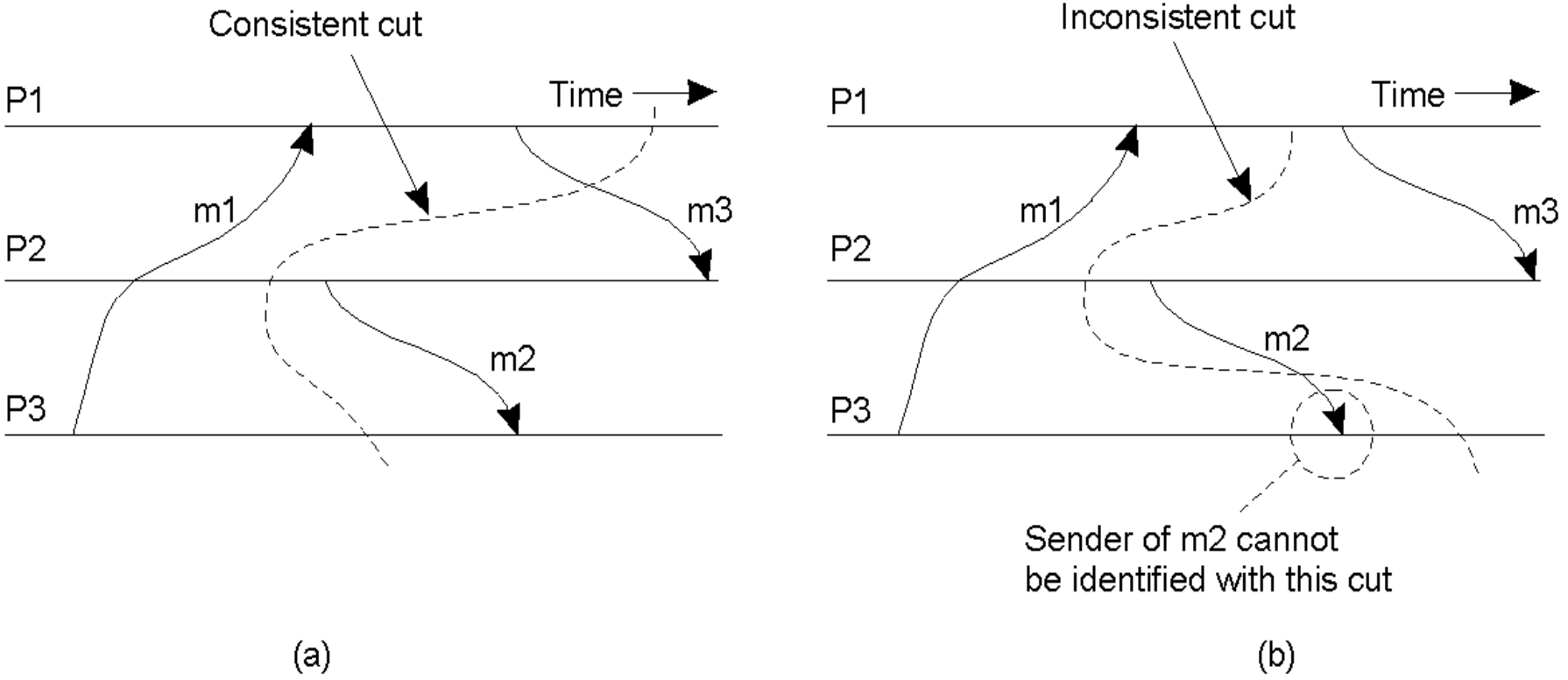
a) Stabil tárolás
b) Összeomlás az egyes lemez frissítése után
c) Blokkhiba

Ellenőrző pontok (1)

- Állapot feljegyzése
- Konzisztens globális állapot

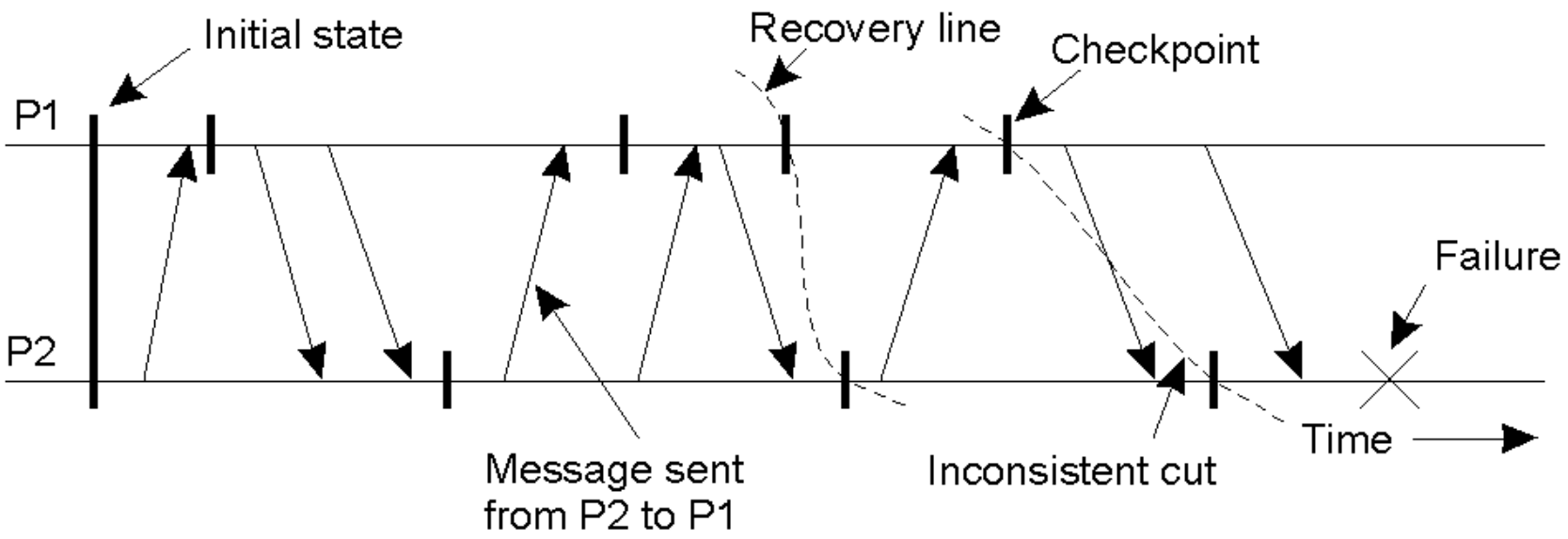
- Felépülési vonal: a legutolsó konzisztens globális állapothoz tér vissza

Ellenőrző pontok (2)



a) Konzisztens metszet
b) Inkonzisztens metszet

Ellenőrző pontok (3)

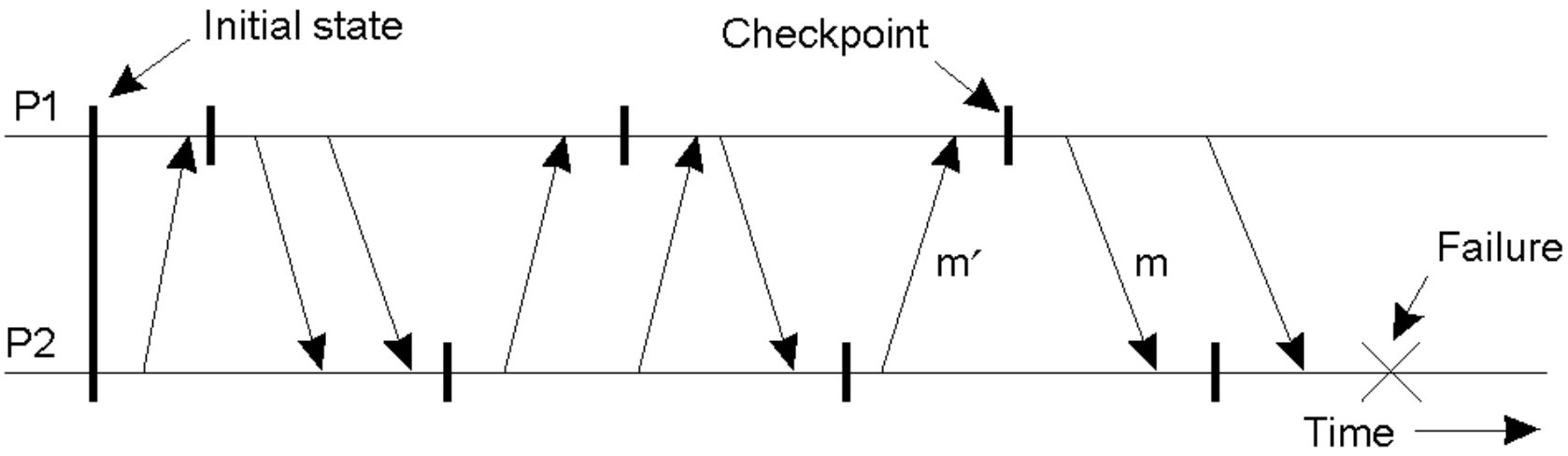


A felépülési vonal.

Független ellenőrző pont készítése (1)

- Minden folyamat időről időre feljegyzi a saját állapotát
- Felépülési vonal megtalálása nehéz – dominóeffektus

Független ellenőrző pont készítése (2)



A dominóeffektus.

Független ellenőrző pont készítése (3)

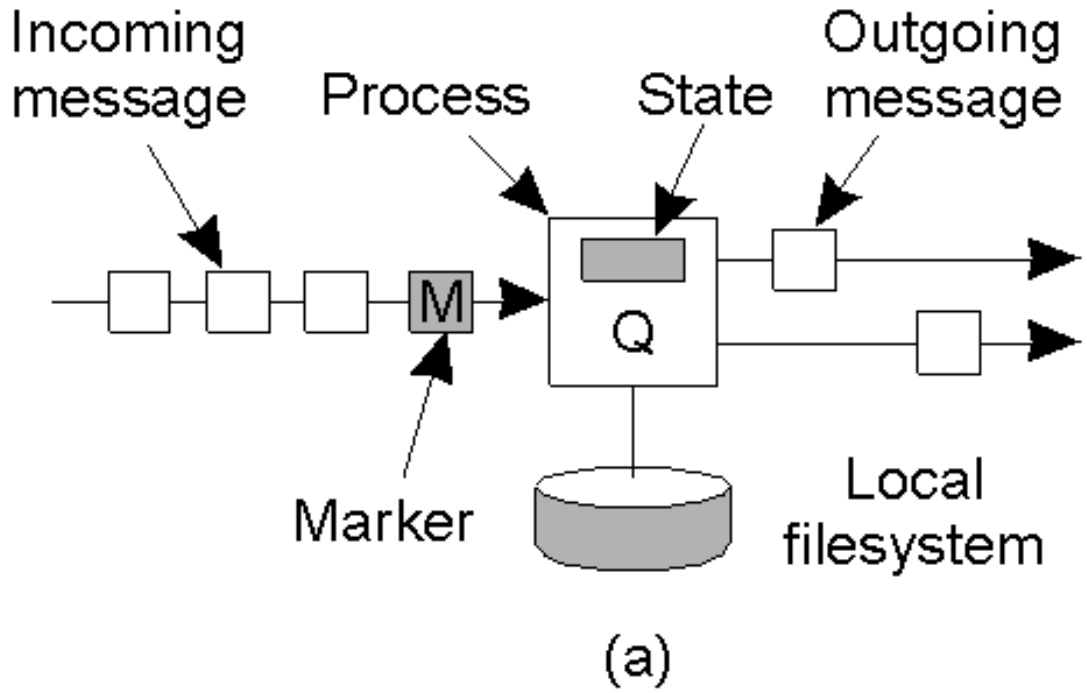
Implementálás – függőségek tárolása:

- CP[i](m): P_i folyamat m-edik ellenőrző pontja
- INT[i](m): CP[i](m-1) és CP[i](m) közötti időintervallum
- P_i folyamat INT[i](m) intervallumban üzenetet küld: (i,m) csatolása
- P_j folyamat INT[j](n) intervallumban üzenetet kap: INT[i](m) -> INT[j](n) függőség feljegyzése

Koordinált ellenőrző pont készítése

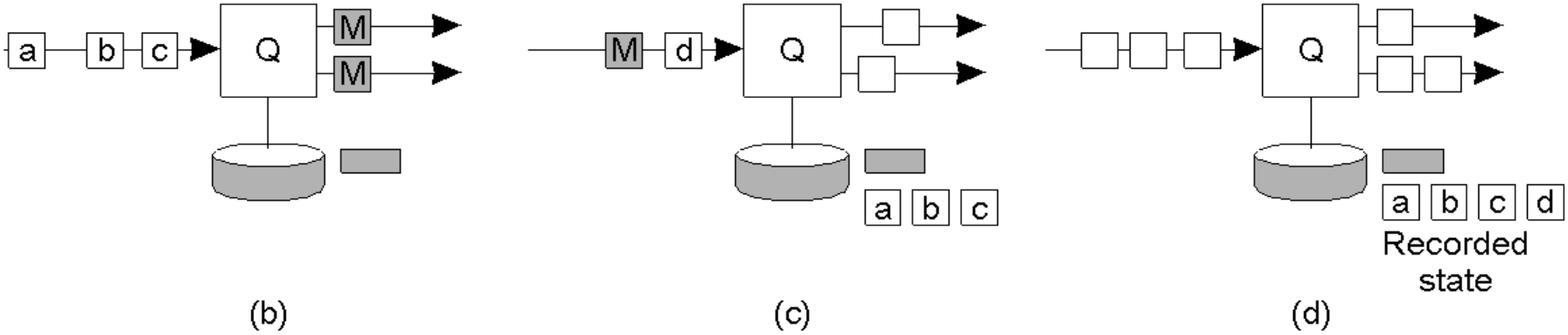
- Minden folyamat szinkronizálódik, és együttesen írja ki a saját helyi állapotát
- Az elmentett állapot automatikusan globálisan konzisztens lesz
- Népszerűbb, mint az előző módszer

Nem blokkoló ellenőrzőpont-koordinálás (1)



a) A folyamat és csatornáinak szerkezete az elosztott pillanatfelvétel elkészítéséhez.

Nem blokkoló ellenőrzőpont-koordinálás (2)



b) Q folyamat a jelzést először kapja meg, ezért feljegyzi saját állapotát.

c) Q valamennyi érkező üzenetét feljegyzi.

d) *Q* megkapja a jelzést a bejövő csatornán és befejezi a csatorna állapotának rögzítését.

Kétfázisú blokkoló ellenőrzőpont-koordinálás

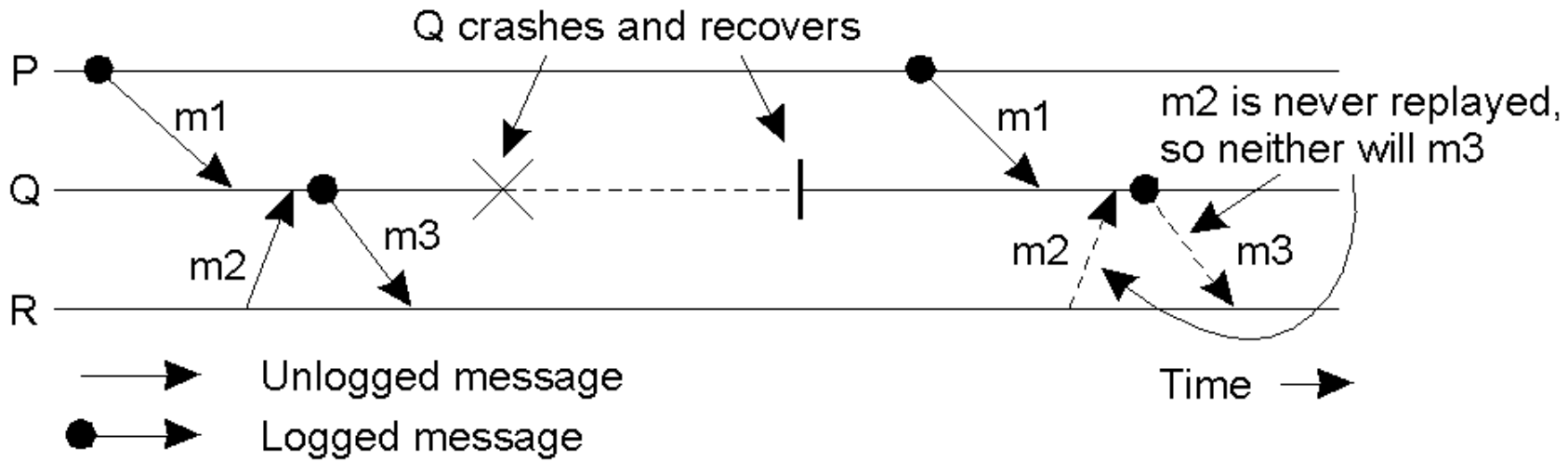
- Koordinátor minden folyamatnak CHECKPOINT-REQUEST üzenetet küld
- Minden folyamat ellenőrző pontot készít és nyugtát küld (a folyamat többi munkája blokkolódik)
- Ha a koordinátor minden nyugtát megkapott: minden folyamatnak CHECKPOINT-DONE üzenetet küld (a folyamatok folytathatják a munkát)

Üzenetek naplózása (1)

- Kiindulási pont: ellenőrző pont
- Üzenetek újrajátszása
- Szakaszos determinisztikus modellben működik jól
- Az üzeneteket mikor naplózzuk?

- Fattyúfolyamat: túléli más folyamatok összeomlását, de állapota nem konzisztens azzal, amelyben az újrainduló folyamat lesz a felépülése után

Üzenetek naplózása (2)



A felépülés után az üzenetek helytelen megismétlése, amely fattyúfolyamathoz vezet.

Üzenetek naplózása (3)

- Üzenet fejlécében sorszám az ismétlések elkerülésére
- Stabil üzenet: többé már nem veszhet el

- Pesszimista naplózóprotokollok
- Optimista naplózóprotokollok