

K2

üzemeltetés

Üzemeltetői Kézikönyv

2.3 verzió

A K2 rendszert az Effice kft. készítette.
E-mail: effice@effice.hu

Kizárólag a Budapesti Értéktőzsde és tagjai számára, belső használatra!

© Copyright Effice Mérnöki Iroda Kft. 1999, 2001.

Ez a dokumentum sem részben, sem egészben nem tárolható információs rendszerekben, nem reprodukálható és nem továbbítható semmilyen formában vagy célból, sem elektronikus, sem fotografikus, sem fotosztatikus, sem mágneses, sem bármi más módon az Effice Mérnöki Iroda Kft. (2083 Solymár, Toldi u. 21, tel: 4290028) előzetes engedélye nélkül.

1.1 verzió MMTS I. 1999.

2.2 verzió MMTS II. 2001.

2.3 verzió BÉT felülvizsgálat, 2002. szeptember

Tartalomjegyzék

| | |
|--|-----------|
| 1. A K2 RENDSZER BEVEZETŐ ISMERTETÉSE..... | 4 |
| 1.1. A K2 RENDSZER FUNKCIÓI..... | 4 |
| 1.2. A K2 RENDSZER FELÉPÍTÉSE | 4 |
| 1.3. A RENDSZER FŐBB RÉSZEI..... | 6 |
| 1.3.1. A megosztott memória | 6 |
| 1.3.2. Megosztott memória táblák..... | 6 |
| 1.3.3. A PGW processz..... | 7 |
| 1.3.4. Az IFSS processz | 9 |
| 1.3.5. Az IFSC | 11 |
| 2. AZ ALAPVETŐ ÜZEMELTETÉSI FELADATOK FAJTÁI..... | 12 |
| 2.1. A LINUX OPERÁCIÓS RENDSZER ÜZEMELTETÉSE..... | 12 |
| 2.2. A K2 RENDSZER INDÍTÁSA | 12 |
| 2.3. A K2 RENDSZER LEÁLLÍTÁSA..... | 12 |
| 2.4. A K2 RENDSZER ÚJRAINDÍTÁSA | 13 |
| 2.5. AZ IFSC KAPCSOLAT ÚJRAÉPÍTÉSE (IFSC RECOVERY) | 13 |
| 3. FUTÁSI PARAMÉTEREK BEÁLLÍTÁSA | 13 |
| 3.1. A PGW FUTTATÁSÁHOZ SZÜKSÉGES KÖRNYEZETI VÁLTOZÓK..... | 14 |
| 3.2. A PGW FUTÁSI PARAMÉTEREINEK BEÁLLÍTÁSA | 14 |
| 3.2.1. Alapértelmezett paraméterek..... | 14 |
| 3.2.2. Konfigurációs file..... | 14 |
| 3.2.3. Parancssor paraméterek | 14 |
| 3.3. AZ IFSS FUTTATÁSÁHOZ SZÜKSÉGES KÖRNYEZETI VÁLTOZÓK | 15 |
| 3.4. AZ IFSS FUTÁSI PARAMÉTEREINEK BEÁLLÍTÁSA | 16 |
| 3.4.1. Alapértelmezett paraméterek..... | 16 |
| 3.4.2. Konfigurációs file..... | 16 |
| 3.4.3. Parancssor paraméterek | 16 |
| 3.5. AZ IFSS FELHASZNÁLÓK ÉS JOGOSULTSÁGOK NYILVÁNTARTÁSA | 16 |
| 3.6. A K2 RENDSZER KONFIGURÁLHATÓSÁGA | 17 |
| 3.6.1. A K2 telepítéskor felvitt konfigurációs fájl..... | 17 |
| 3.6.2. A K2 felhasználó által nem módosítható konfigurációs paraméterek | 17 |
| 3.6.3. A BÉT által használt LINUX/UNIX felhasználói azonosító és annak jogosultságai | 17 |
| 4. A K2 RENDSZER MŰKÖDÉSÉNEK NYOMON KÖVETÉSE - HIBAKERESÉS | 17 |
| 4.1. A PGW ÉS AZ IFSS PROCESSZEK MEGFIGYELHETŐSÉGE..... | 17 |
| 4.2. A RECOVERY LOG FÁJL TARTALMA | 18 |
| 4.3. A MEGOSZTOTT MEMÓRIA TARTALMÁNAK KIÍRATÁSA | 20 |
| 4.4. A K2 RENDSZER ÁLLAPOTÁNAK MONITOROZÁSA | 20 |
| 4.5. A MONITOROZÁS EREDMÉNYÉNEK RÖGZÍTÉSE | 22 |
| 4.6. A PGW ÁLTAL HASZNÁLT ERŐFORRÁSOK MEGJELENÍTÉSE | 22 |
| 4.7. ERŐFORRÁS FELSZABADÍTÁS A PGW VAGY IFSS HIBÁS KILÉPÉSE UTÁN | 23 |
| 4.8. K2 FELHASZNÁLÓI JELSZÓ MÓDOSÍTÁSA..... | 23 |
| 5. A K2 ÉS AZ IFSC EGYÜTTMŰKÖDÉSÉT BEMUTATÓ DEMO ÉS TESZT PROGRAMOK ÜZEMELTETÉSE..... | 23 |
| 5.1. GET_TABLE | 23 |
| 5.2. GET_OB | 24 |
| 5.3. OB_WATCH..... | 25 |
| 5.4. SEND_ORDER..... | 25 |
| 5.5. GEN_ORDER | 26 |
| 5.6. DEMO..... | 26 |
| 5.7. ADATFORMÁTUM ÁTALAKÍTÓ AWK PROGRAMOK | 27 |
| 5.7.1. get_ob.awk | 27 |
| 5.7.2. get_table.awk | 27 |
| 5.7.3. pobo.awk | 27 |
| 5.8. IFSTEST: FELHASZNÁLÓI PROGRAMOK TESZTELÉSE | 28 |

TARTALOMJEGYZÉK

| | |
|---|-----------|
| 6. FÜGGELÉK A: HIBAÜZENETEK..... | 28 |
|---|-----------|

1. A K2 rendszer bevezető ismertetése

1.1. A K2 rendszer funkciói

A K2 (Kapcsolati Kiszolgáló) nagy kapacitású valós idejű kapcsolatot tesz lehetővé az MMTS¹ kereskedési rendszer és az ehhez illesztett felhasználói rendszerek között.

A K2 két azonosan felépített rendszerből áll, a K2_a és K2_d alrendszerekből. Az első az azonnali kereskedési rendszerrel (MMTS I.) a második a határidős és opciós rendszerrel (MMTS II.) biztosítja a kapcsolatot. A két rendszer teljesen azonos felépítésű, eltérések a memória táblák mezőiben, és az üzenetkezelésben (ASN.1) mutatkoznak. Ezért a továbbiakban K2 alatt a két alrendszer bármelyikét is értjük, abban az esetben, ha *nincs különbség* a két alrendszer között a tárgyalt tulajdonság szempontjából.

A K2 lehetővé teszi, hogy a többi kereskedési munkaállomás használata mellett közvetlen valós idejű adatáramlás történjen az MMTS és a csatolt rendszer közt, így például a bróker rendszer azonnal értesüljön az MMTS-ben lévő minden ajánlatról és kötésről, és a bróker rendszerben összegyűjtött és ellenőrzött megbízások azonnal eljussanak a kereskedési rendszerbe. Mivel a K2 kialakítása olyan, hogy az MMTS központi rendszerrel azonos sebességi és teljesítmény paraméterekkel rendelkezik, lehetőség van arra is, hogy segítségével online kereskedési funkciók is a bróker rendszeren keresztül valósuljanak meg a kereskedési munkaállomások helyett vagy mellett. A K2 felhasználása mellett lehetőség van a hagyományos munkaállomások további alkalmazására, de arra is, hogy azok szerepét akár részben, akár egészben átvegye a bróker rendszer.

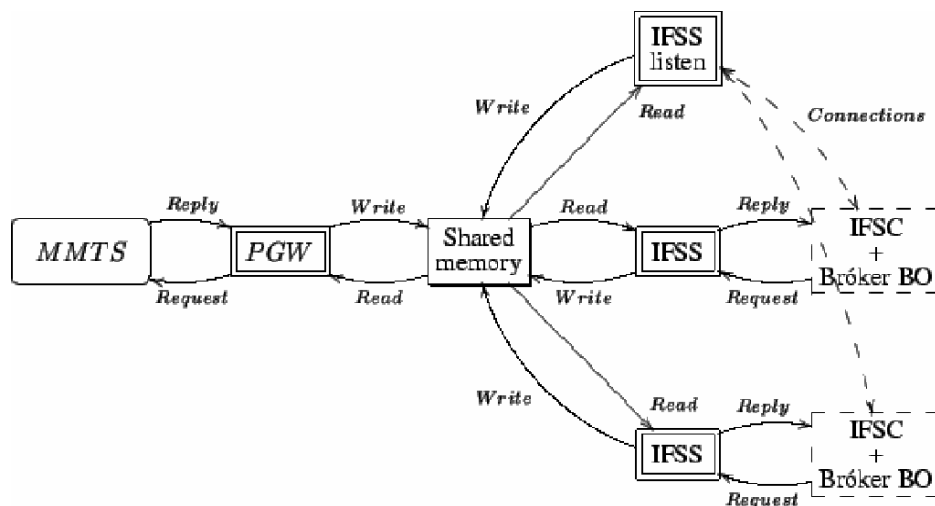
A K2 egy nagy megbízhatóságú, robusztus alkalmazás, mely sokkal nagyobb sebességgel és megbízhatósággal képes ellátni feladatát, mint az eddig alkalmazott, vagy tervezett, munkaállomáson alapuló megoldások.

A K2 fontosabb felhasználási területei

- Kereskedési információ valós idejű letöltése és frissítése
- Előkészített ajánlatok bevitele
- Ajánlatok törlése, módosítása
- Triggerelt ajánlatbevétel, törlés, módosítás
- Automatikus kereskedés
- Saját kereskedési rendszer illesztése
- Egyéni kereskedési munkaállomások létrehozása
- Piaci információk valós idejű feldolgozása, kereskedő támogatás
- Riasztási funkciók, beavatkozás
- Kereskedés felügyeleti és vezetési funkciók

1.2. A K2 rendszer felépítése

¹ Multi-Market Trading System – a Budapesti Értéktőzsde (BÉT) kereskedési rendszere beleértve mind az azonnali, mind pedig a határidős és opciós piacokat kiszolgáló rendszert.



ábra 1.1: A K2_a és K2_d felépítése

A rendszer felépítését az 1.1.² ábra szemlélteti. Az MMTS kereskedési rendszer felé a kapcsolatot a PGW process, a felhasználói rendszerek felé pedig az IFSS process biztosítja az IFSC library segítségével. A két process megosztott (shared) memória és szemafor segítségével tart kapcsolatot.

PGW (polling gateway) Az MMTS munkaállomást helyettesítő program, amely az MMTS-felől nézve ugyanúgy működik, mint a munkaállomás. Azonban az adatokat nem ablakokban jeleníti meg, hanem a megosztott memóriában tárolja le.

IFSS (interface socket server) Kiszolgáló program, amely a felhasználói rendszereket szolgálja ki hálózati kapcsolaton keresztül az osztott memória táblák olvasása és írása által.

IFSC (interface socket client) Kliens könyvtár, amelynek segítségével más rendszerek tudnak kapcsolódni a K2-höz. Az IFSC a kapcsolódó rendszer gépén fut, és könyvtári függvények formájában nyújt felületet a kapcsolódáshoz.

A K2 mind az MMTS rendszer felé, mind a felhasználói rendszerek felé hálózati protokollok segítségével kapcsolódik. MMTS felé egy TSMR³ nevű üzenet közvetítőt használ, ami TCP/IP és UDP/IP-t igényel. Az IFSS standard TCP/IP socket alapú.

A K2-n átmenő tranzakcióknak alapvetően két fajtája van:

lekérdezés PGW process periodikusan lekérdezi a kereskedési adatok változásait és beírja a megosztott memóriába. Az IFSS process pedig a hozzáérkező lekérdezésekre válaszul elküldi a változásokat.

ajánlatbeviteli tranzakció Ajánlatbevitel (új ajánlat elküldése, meglévő módosítása vagy törlése) során az IFSS vesz egy ilyen kérést és beszúrja a megosztott memóriába. A PGW a lekérdezési ciklus végén megnézi van-e új ajánlatbeviteli rekord és elküldi azt az MMTS-be. Az MMTS általadott választ megosztott memóriába írja. A felhasználó a megfelelő *lekérdezés* segítségével értesülhet *tranzakciójának* állapotáról.

² Az ábra mind a K2_a-ra, mind a K2_d-re külön-külön érvényes.

³ Figyelem: az MMTS I. és MMTS II. rendszerek – így a K2_a és K2_d is - különböző verziószámú TSMR- t használnak!

1.3. A rendszer főbb részei

1.3.1. A megosztott memória

A megosztott memória a K2 lényeges eleme. Tulajdonképpen ez biztosítja a processzek közötti adatcserét.

A memóriakezelés (létrehozás, kapcsolódás stb.) *System V IPC* függvények hívásával történik. A memória hozzáférés szinkronizálását egy szemafor biztosítja.

A megosztott memóriát a PGW hozza létre miután bejelentkezett az MMTS-be. A memória mérete igazodik az MMTS memória táblaméret paramétereire¹, mivel méretét az MMTS-ből kapott adatok alapján határozza meg a PGW. A PGW a megosztott memória létrehozása után felkapcsolódik arra és TSMR lekérdezések segítségével feltölti azt. Az IFSS a már létrehozott megosztott memóriára kapcsolódik fel. A megosztott memóriát a programok (PGW, IFSS) nem szüntetik meg. Megszüntetni a `ripcc` eszközzel lehet, amely az `ipccrm` operációs rendszer parancson alapul. (lásd 10. fejezet).

A megosztott memóriához a PGW process és az IFSS processzek csatlakoznak állandó jelleggel. Alkalmassint természetesen a `dump` és `MEM` eszközök is hozzá kapcsolódnak. A csatlakozó processzek a megosztott memória olvasásának és írásának idejére a szemafor segítségével kizárólagos hozzáférést kérnek.

1.3.2. Megosztott memória táblák

A megosztott memóriában az alábbi táblák találhatók:

1. Piacok tábla.
2. Instrumentumok tábla.
3. Iparágak tábla.
4. Kereskedési táblák (Részvény, Állampapír, stb.) tábla.
5. Értékpapírok tábla.
6. Felhasználók tábla.
7. Brókercégek tábla.
8. Ajánlatok tábla.
9. Kötések tábla.
10. Fix ügyletek tábla.
11. Ajánlati könyvek².
12. Ajánlatbevitel tábla³.
13. Ajánlatbeviteli hibüzenetek tábla⁴

¹ Az ajánlatbevitelhez kapcsolódó táblák méreteit konfigurációs file határozza meg.

² Az ajánlati könyvek kezelése a többi táblától eltérően történik, valamint jelentősen eltér egymástól a K2_a és K2_d esetében.

³ Ez nem MMTS-ből származó adatokat tartalmaz, hanem a felhasználók által az MMTS-be bevitt ajánlatbevitelt.

A fenti táblák mellett, amelyek a kereskedéssel kapcsolatos adatokat tartalmazzák, más a rendszer működésével kapcsolatos adatok is megtalálhatók a megosztott memóriában, amire mind a PGW és mind az IFSS processzeknek szüksége van (pl.: tábla méretek és címek, legnagyobb ajánlatbeviteli azonosító, rendszerazonosító.)

Az egyes táblák egyforma méretű rekordokból állnak és az ajánlati könyvek kivételével előre definiált kulcs szerint sorrendezettek. A rekord változások követését a változásszámok teszik lehetővé. Ajánlati könyvek kivételével minden táblához tartozik egy változásszám-rekordszám párost tartalmazó változásszám tábla, változásszám szerint rendezve. Ha új rekord kerül beszúrára, akkor az megkapja a legnagyobb változásszámot és változásszám táblához új sor fűződik hozzá. Ha egy meglévő rekord kerül frissítésre, akkor szintén megkapja a legnagyobb változásszámot és a változásszámtábla korábbi rekordja után a következő rekordok eggyel előbbre mozdítódnak (az eredeti rekord törlésre kerül) és a végére egy új kerül beszúrára. Ezáltal a lekérdezések során csak a legutolsó változásszámot kell megjegyezni és a következő lekérdezés során a változásszám táblából a legutolsó változásszámot követő rekordokat kiolvasni, hogy megkapjuk az új vagy megváltozott rekordokat.

A K2_a-ban az ajánlati könyvek tábla egy rekordja egy értékpapír ajánlati könyvét tartalmazza (Jelenleg az MMTS által megszabott 20 mélységig). Összes értékpapír ajánlati könyvének figyelésére nincs mód, mert az MMTS nem tudná azt kiszolgálni. Ezért előre meghatározott számú ajánlati könyv figyelése lehetséges, vagy egy lekérdezési ciklusban csak korlátozott számú ajánlati könyv kerül lekérdezésre, de ebben az esetben egyes ajánlati könyvek nem minden ciklusban frissítődnek. A felhasználói rendszerek az IFSS-en keresztül meghatározhatják, mely értékpapírok ajánlati könyveit figyelje a PGW.

A K2_d-ben azonban más megoldást kellett alkalmazni az MMTS II. eltérő viselkedése miatt. Az MMTS II. rendszer az ajánlati könyvet teljes hosszában küldi, ezért a K2_d-ben egy külön *ajánlati könyv* tábla szolgál az adatok tárolására. Az ajánlati könyv tábla méretét a konfigurációs file-ban lévő `ordlistsize` paraméterrel lehet megadni. Az *ajánlati könyv kibontva* „tábla” tulajdonképpen az ajánlati könyv tábla egy nézete.

Az *értékpapír tábla* adatait nem csak rekordszintű változással lehet követni, hanem mezőszintű változások szerint is. Az *értékpapír tábla* mezői két csoportba oszthatóak: az egyik részük statikus, amik az indulást követően csak egyszer kerülnek letöltésre. Minden rekord sorszámot kap (`idx`), amely nullától folyamatosan növekvő. A statikus adatok az IFSC fizikai sorrendű rekordlekérdezésével tölthetők le, amelyek a sorszámokra hivatkoznak. A többi mező napközben folyamatosan is változhat. A PGW egy *rekord x mező* méretű mátrixban tartja nyilván a mezőváltozás számokat. Ha egy rekord egyik mezője megváltozik, akkor a mátrix megfelelő eleme új, az eddigi legnagyobb változás számot kapja meg. Az IFSC mezőváltozás lekérdezése ezekre a változás számokra hivatkozik és a válasz azokat a mezőkód-érték párokat tartalmazza, amelyek változás száma nagyobb a kérdés változás számánál.

1.3.3. A PGW processz

A PGW process az MMTS rendszer egy kliens programja. Hasonlóan a kereskedői munkaadomáshoz bejelentkezik és kéréseket küld az MMTS rendszerbe, ahonnan minden esetben választ kap. Ezeket az üzeneteket természetesen a TSMR-en keresztül kezeli, ezért a TSMR léte¹ – legalábbis a kliens oldal – alapkövetelmény a PGW számára. A PGW általa kezelt kereskedési adatokat a megosztott memóriába (lásd 2. fejezet) tárolja, amit más programok is elérhetnek. A K2_a és K2_d rendszerek eltérő PGW-eket használnak, ezeket `pgw1`-nek és `pgw2`-nek nevezzük. Működésük külső szemlélő szempontjából megegyező.

⁴ Ez a tábla az ajánlatbevitel tábla kiegészítése, külön nem kérdezhető le.

¹ Ebből következően a SYSSRV is szükséges!

Az MMTS rendszerbe küldött kéréseknek két fajtája van:

lekérdezések Az előző lekérdezés óta keletkezett új vagy megváltozott adatok frissítése. Egy lekérdezés mindig egy adott adattípusra vonatkozik. (ajánlat, értékpapír, egy adott értékpapír ajánlati könyve . . .)

ajánlatbeviteli kérések Új ajánlat bevitele vagy meglévő módosítása, illetve törlése. Válaszként csak a tranzakció sikere vagy sikertelensége érkezik a felhasználónak szánt üzenettel együtt². Sikeres új ajánlat bevétel esetén az üzenet tartalmazza, az új ajánlat ajánlatszámát, amit a PGW kikeres és visszaír az ajánlatbeviteli rekordba.

Sikertelen üzenet esetén az érkező hibaüzenetet beszúrja a hibaüzenet táblába. A bevitt ajánlatot, illetve azok megváltozásait, az ajánlatokra vonatkozó megfelelő *lekérdezéssel* lehet megkapni.

A PGW TSMR segítségével csatlakozik az MMTS-hez, mint minden más felhasználó. A PGW a sikeres bejelentkezést követően periodikusan lekérdezéseket küld az MMTS-be. A sikeres bejelentkezést követő első lekérdezés az MMTS u.n. *stats* tömbjét kérdezi le, ami az MMTS memóriatábláinak pillanatnyi méretét és más globális MMTS adatokat tartalmaz. Ezen adatok és a konfigurációs file alapján a PGW létrehozza a megosztott memória táblákat. Ezt követően azoknak a tábláknak az alapadatai kérdeződnek le, amely táblákban a rekordok száma állandó. (*market*, *instr*, *board*, *sector*, *secboard*, *firm* és *user*) Ezután veszi kezdetét a tulajdonképpeni periodikus működés. A periodikus működés során az alábbi táblákra vonatkozó *változás* lekérdezéseket eszközli a PGW: *market*, *instr*, *board*, *firm*, *user* és *secboard*. Ezekben csak létező rekordokra vonatkozó változás lehet, új rekordra vonatkozó adat sohasem érkezik. Az *order*, *negdeal* és *trade* táblákra csak a periodikus szakaszban történik lekérdezés. Ezekből a lekérdezésekből, mind már létező rekordra vonatkozó változás, mind új adatrekord származhat. Különösen igaz ez a *trade* táblára, ahol egy extra lekérdezés (*last_n_trades*) is szerepel annak érdekében, hogy a PGW az összes kötésről szerezzen tudomást. Minden lekérdezési ciklus végén a konfigurációs file-ban megadott ideig *sleep* segítségével kiegészíti a ciklusidőt.

Kissé más módon történik az ajánlati könyvek lekérdezése. Míg a K2_a PGW-je külön listában tartja nyilván a *market_byorder* és *market_byprice* ajánlati könyvek sorát, addig a K2_d egy ajánlati könyv táblában tartja az adatokat és ebből képi a *market_byorder* és *market_byprice* ajánlati könyveket. A PGW a pillanatnyi konfiguráció szerint beállított ajánlati könyveket kérdezi le.

Ajánlatok beviteli, módosítási és törlési kéréseket is továbbítja a PGW az MMTS-hez. Ezeket egy az IFSS által írt megosztott memória táblában keresi. Egyenként továbbítja őket az MMTS-be, a visszakapott státuszt, esetleg ajánlatszámot vagy hibaüzenetet pedig bejegyz a táblába. Az IFSS *SIGUSR1 signal*-t küldhet a PGW-nek, ha új rekordot írt az ajánlatbeviteli táblába, ezáltal gyorsítva a rendszer működését. A PGW a *signal* vétele esetén felfüggeszti az aktuális *sleep*-et és MMTS-be elküldi az új ajánlatbeviteli rekordot vagy rekordokat. Ez azonban csak akkor lehetséges, ha a konfigurációs file-ban a *sendsignal* paraméter értéke nem nulla.

A megosztott memóriát olvasás és írás alatt PGW a szemaforral védi. A szemafor csak a periodikus működés elkezdésekor állítja először szabad állapotba. A periodikus működés során, a lekérdezések és ajánlatkérések elküldése alatt a szemafor lezárt állapotba helyezi. A *freeseenabled* konfigurációs paraméter beállítása esetén a TSMR üzenetváltásokra azonban a szemafor felengedi, hogy kommunikációs hibák miatti várakozás alatt is lehessen a megosztott memóriából információ kinyerni (pl. a PGW process állapotáról).

Az MMTS II. rendszer, első bejelentkezésétől az első sikeres jelszó változtatásig az MMTS felhasználónak korlátozott funkciókat (csak lekérdezéseket) engedélyez. Ezért ilyen esetben a PGW process figyelmeztető üzenetet ír ki. Ekkor érdemes a K2-t leállítani és a jelszót *k2changepwd* szkripttel megváltoztatni. A jelszó változtatást a biztonság érdekében az MMTS I rendszer esetén is végre kell hajtani az éles rendszerbe történő első bejelentkezéskor a BÉT-től kapott jelszó lecserélése érdekében.

² A felhasználónak megfelelő nyelvű üzenet érkezik.

1.3.4. Az IFSS processz

Az IFSS (interface socket server) process a K2 kiszolgáló szervere felhasználói rendszerek részére. TCP/IP socket kapcsolaton keresztül szolgálja ki a lekérdezéseket, illetve az ajánlatbeviteli kéréseket. A kereskedési adatokat a PGW által feltöltött megosztott memóriából veszi és az ajánlatbeviteli kéréseket is oda írja be. A K2_a és K2_d különböző IFSS-eket használ (*ifss1* és *ifss2*), amelyek a külső szemlélő számára egyenértékűek.

IFSS működése a következő fázisokból áll:

1. kapcsolatra váró szerver process indulása
2. kapcsolat felvétel - kiszolgáló process lehasítása
3. üzenet váltások
4. kapcsolat bontás

Indulás során felolvassa a konfigurációs file-t, IFSS felhasználói adatokat, indítási kapcsolókat, megnyitja a log file-okat, önállóan futó processzbe másolja magát, az eredeti shell-ben futó program kilép. Az önálló process legyártja azt a socket-et, ahol figyel, majd várja a kapcsolódásokat.

Kapcsolat felvétel esetén a kliens a figyelő socket-re kapcsolódik, az önállóan futó IFSS process észleli a felkapcsolódást és lehasít magából egy kiszolgáló processzt az adott kliens kiszolgálására. A kiszolgáló process felkapcsolódik a megosztott memóriára és elkezd egy ciklust, amiben a socket-re küldött üzeneteket olvas és válaszol azokra. Legelőször bejelentkezési üzenetet vár.

Minden üzenetnek típusa van, amely az üzenet elején található fejlécben található. Az üzenet típusa meghatározza az üzenet adatainak formáját. Az üzenetben az adatok nullával lezárt string formában vannak ábrázolva. Az adatok típusa előre meghatározott.

A kiszolgáló process a socket-jén egy kérő üzenetet vár, ha az üzenet minden karaktere megérkezett, akkor értelmezi annak típusát és a megfelelő eljárásnak adja át. Az adott eljárások a kérés adatait értelmezik és összeállítanak egy válasz üzenetet, amit a socket-en keresztül elküldenek, majd ismét várakoznak a következő üzenetre.

Lekérdezések esetén a hatékonyság növelése érdekében a válasz üzeneteket (amik egy-egy rekordnak megfelelő adatokat tartalmaznak) egy csomag üzenetbe rakja össze. A kliens a csomagot szétbontja és az első üzenetet odaadja a hívónak. Mindaddig, míg a hívó program a következő rekordot kéri a kliens nem küld lekérdező üzenetet, hanem a csomag következő üzenetét veszi elő. Ha a csomag elfogyott, akkor küld egy lekérdező üzenetet. Ha a hívó program a rekordokat egymás után dolgozza fel, akkor ez az összecsomagolás egy nagyságrenddel megnöveli a többszörös lekérdezés sebességét.

Az üzenetek típusai a következők:¹

Ki- és bejelentkezés: Bejelentkezés (IFS_LOGIN_REQ) hatására a kiszolgáló process ellenőrzi a felhasználó nevét és jelszavát, státuszát. Ha minden rendben van, akkor a további üzeneteket a felhasználó jogosultságai szerint kezeli. Ha a felhasználói adatok nem megfelelőek, vagy a felhasználó felfüggesztett állapotú, akkor hibaüzenet küldése után a socket kapcsolatot lebontja és befejezi a működést. Válaszüzenetben a rendszer indításának azonosítója is szerepel, aminek segítségével el lehet dönteni, hogy a korábbi lekérdezések érvényben vannak-e.

¹ A pontos adatformátumot az *ifs_msg.h* file tartalmazza.

Kijelentkezés (IFS_LOGOUT_REQ) esetén választ küld és lezárja a socket kapcsolatot és kilép a kiszolgáló process.

Lekérdezés: Adott tábla rekord változásának lekérdezése (IFS_SEQ_TABLE_REQ) változás szám alapján. Minden táblára ki lehet adni kivéve ajánlati könyv. Hatására a kiszolgáló process kikeresi az adott tábla adott változásszámot követő rekordját és onnan kezdve annyi rekordról készít válasz üzeneteket amennyi egy csomagba elfér, válaszul ezt a csomagot küldi. Minden üzenetben az adott rekord változás száma is fel van tüntetve, aminek a segítségével a következő rekord lekérdezhető.

Értékpapír tábla esetén mező szintű változás figyelésre is mód van. Az értékpapír tábla statikus mezőit a fizikai sorrendű rekord lekérdezéssel (IFS_IDX_TABLE_REQ) lehet lekérdezni. A többi folyamatosan változó mezőket a mező változás (IFS_FIELD_CHG_REQ) kérés segítségével lehet követni. Mind két lekérdezés a rekord változás lekérdezéshez hasonlóan csomagokat alkalmaz a lekérdezés üzenetváltások gyorsítására.

Ajánlati könyv kezelés: Adott értékpapírtáblához tartozó ajánlati könyvet először fel kell venni a figyelési listára az ajánlati könyv felvétele (IFS_SET_OB_CONF_REQ) üzenettel. Hatására a kiszolgáló process felveszi az adott értékpapírtáblát a figyelési listára, ha az még nem volt felvéve és van még szabad hely.

A figyelt értékpapírtáblákat az ajánlati könyv lista (IFS_OB_LIST_REQ) üzenettel lehet lekérdezni. Hatására a kiszolgáló process elküldi a figyelt értékpapírtáblák azonosítóit.

Adott értékpapírtáblához tartozó ajánlati könyvet IFS_ORDERBOOK_REQ üzenettel lehet lekérdezni. Hatására a kiszolgáló process egy üzenetben elküldi a teljes ajánlati könyvet. Az ajánlati könyv maximális mélységét az MMTS rendszer határozza meg, jelenleg ez 20. A kiszolgáló process változó hosszúságú üzenettel válaszol az adott mélységszámtól függően.

Ajánlatbeviteli kérés: Ajánlatbeviteli kérés (IFS_ORDERENTRY_REQ) hatására a kiszolgáló process új rekordot szúr be az orderentry táblába, egyedi ajánlatbeviteli azonosítót ad neki és a változás számot is frissíti. Ez után az ajánlat adattartalmán egyszerű formai ellenőrzést végez és az ellenőrzés eredményétől függően beállítja az ajánlat státuszát. Végül válasz üzenetet küld, amely tartalmazza az ajánlatbeviteli kérés azonosítóját. Az üzenet adat tartalma függ attól, hogy új ajánlat bevitelét vagy meglévő módosítását, ill. törlését kérték-e.

Ajánlatbeviteli kérés státusváltás kérés (IFS_ORDERENTRY_STC_REQ) hatására a kiszolgáló process ellenőrzi az eredeti és a kért státusz, majd ha azok megfelelőek az adott azonosítójú rekord státuszát megváltoztatja.

Mindkét esetben az IFSS *SIGUSR1 signal*-t küldhet a PGW-nek gyorsabb (azonnali) küldésre ösztökélve azt.

Kapcsolat bontása: Kijelentkezés hatására a kiszolgáló process lebontja a socket kapcsolatot és kilép. Ha a kapcsolatban hiba lép fel (vagy kommunikációs, vagy a kliens program kilép) akkor a socket olvasás hibával tér vissza. Ekkor a kiszolgáló process szintén kilép.

Felhasználók

A K2-ben kétféle felhasználó fogalmat különböztetünk meg:

MMTS user A PGW process elindításakor egy előre megadott MMTS felhasználó (MMTS user) nevében jelentkezik be az MMTS rendszerbe. A lekérdezéseket és az ajánlatbeviteli tranzakciókat ennek az MMTS felhasználó nevében kezdeményezi, az MMTS a jogosultságokat ennek megfelelően kezeli.

IFS user A felhasználói rendszer amikor felveszi a kapcsolatot a K2/IFSS-sel, akkor megad egy IFS felhasználó nevet és jelszót. Az IFSS ennek alapján kezeli a további kéréseket. Minden felhasználónévhez jogosultságok tartoznak. Ezek a felhasználóneveket csak az IFSS használja, az MMTS rendszerbe nem továbbítja.

Az IFSS felhasználók jogosultságaival kapcsolatos részletek a 3.5. pontban találhatók (16. oldal).

1.3.5. Az IFSC

Az IFSC olyan programozói könyvtárba foglalt függvényekből áll, amely a függvényhívásokban megadott paraméterek segítségével kérő üzeneteket állít össze, amit az IFSS részére elküld, majd a válasz üzeneteket vesz és azok adatait a függvények paraméterében visszaadott buffer-ben adja át. Az K2_a és K2_d alrendszerekben *ugyanaz* az IFSC működik! A függvények a következő közös információkat tárolnak a kapcsolatileírókban változóiban:

Rekordváltozások változásszáma Rekordváltozások utolsó lekérdezésben visszaadott változásszáma táblánként.

Fizikai rekord index Legutolsó fizikai sorrendű rekord lekérdezés index-e táblánként.

Mező változásszám Mező változás utolsó változás száma táblánként.¹

Ajánlati könyvek maximális száma A felvehető ajánlati könyv figyelések maximális száma, amit bejelentkezéskor az IFSS megküld.

Ajánlati könyvek változásszáma Az ajánlati könyvek utolsó változásszáma ajánlati könyvenként.

Vételi buffer Utolsó vett üzenet.

Küldési buffer Utolsó elküldött kérés.

Nyomkövetés Nyomkövetési mód.

Nyomkövetési file Nyomkövetési file leírója.

Az IFSC függvények a lekérdezés eredményét a vételi bufferben tárolja és a buffer pointer adja vissza. A felhasználói programnak a következő függvényhívás előtt az adatokat el kell másolnia, mert azok a következő üzenet vétele során felülíródnak. Ajánlatbeviteli kérések feladásánál adatokat bufferben kapja az IFSC függvény, ahonnan az adatokat elmásolja. A bufferben lévő adatok nullával elválasztott fix hosszúságú karakter sorozatok.² Az adatok program változóba történő szétválasztás, illetve változókból történő összeállításhoz az IFSC segédfüggvények nyújtanak segítséget.

Az IFSC függvényekről és segédfüggvényekről külön „K2/ifsc Felhasználói kézikönyv” tartalmaz leírásokat. A felhasználói programokba a következő deklarációs file-okat kell hivatkozni:

ifsapi.h IFSC függvények deklarációja.

ifsdefs.h IFSC konstansok definíciója.

ifsutil.h IFSC segédfüggvények deklarációja.

fields.h Értékpapír tábla mező kódok.

A deklarációs header file-ok C nyelven lettek megírva. A c2p.awk-val lehetőség van PASCAL formátumra alakításra.

Unix platformon a felhasználói programokat a libifs.sl (libifs.so) megosztott könyvtárral kell linkelni. Windows NT platformon az ifsc.dll-t kell használni. Egyes fordítók megkövetelik az ifs.lib statikus könyvtárat is, amely az ifs.dll-ben megadott függvényeket tölti be. Az ifs.lib-et mindig az adott fordító fejlesztési környezetében megtalálható eszközök segítségével kell előállítani.

¹ Jelenleg csak értékpapír táblára lehet ilyet kérdezni.

² Az adatokban is lehetnek null karakterek!

2. Az alapvető üzemeltetési feladatok fajtái

2.1. A Linux operációs rendszer üzemeltetése

A Linux üzemeltetési feladatok részletesebb leírásával ebben a kézikönyvben nem foglalkozunk. Az operációs rendszer üzemeltetésével kapcsolatban a BÉT a folyamatos üzemet, a naplózási területek időnkénti mentését és/vagy törlését, a teljes rendszerről pedig kéthetente, de a szoftver disztribúciót követően mindenképpen biztonsági másolat készítését várja el.

2.2. A K2 rendszer indítása

A PGW csak akkor működik, amikor az MMTS-sel kapcsolatot tud tartani, azaz minden reggel el kell indítani és délután a kereskedés végén le kell állítani. Minden MMTS indítás után – mivel a korábbi adatok érvényüket veszítik – a K2-t üres adatokkal kell elindítani.

A „k2start 1” vagy „k2start 2” parancs a PGW és IFSS processzeket elindítja.

A K2 elindítani vagy újraindítani a következő paranccsal lehet:

```
k2start 1 [options]
```

vagy

```
k2start 2 [options]
```

Az elindítást követően, ha a jelszó nem lett megadva a parancssorban, a PGW megkérdezi azt. Ha nem volt az előtérben futás megadva, akkor a jelszó bekérését követően háttérben futó processzt választ le magáról és a szülő program befejezi működését. Ha az előtérben futás meg lett adva, akkor a k2start az IFSS processzt *nem* indítja el!

Az indítás folyamata:

PGW indítása létrehozza szemafor, bejelentkezik az MMTS-be, létrehozza a megosztott memóriát és lekérdezések által feltölti azt. A *watchpgwenabled* konfigurációs paramétertől függően a PGW két processzre válik szét. A gyerek processz ezek után periodikusan kérdezi le a változásokat és elküldi az ajánlatbeviteli kéréseket, a szülő processz pedig a gyerek processz állapotát figyeli.

IFSS indítása A PGW elindítása után 5 másodperccel a k2start az IFSS processzt is elindítja. Mihelyt a PGW felengedi a szemafor az IFSS képes a kliensek kiszolgálására.

2.3. A K2 rendszer leállítása

A PGW-t és IFSS-t leállítani a k2stop 1 vagy k2stop 2 paranccsal lehet:

```
k2stop 1
```

vagy

```
k2stop 2
```

A leállítás folyamata:

A `k2stop` szkript a `ps` Unix parancs segítségével kikeresi a megfelelő PGW és IFSS processzt és `SIGTERM` signal-t küld azoknak. A PGW `SIGTERM` signal vétele során, kijelentkezik az MMTS-ből és terminálja a processzt. A processzek leállása után a megosztott memóriát és a szemafor is letörli.

Önállóan PGW processzt, az Unix `kill` parancsával lehet leállítani. Ha a PGW leáll, akkor nem törli le a megosztott memóriát és a szemafor, annak olvasására a felhasználói rendszereknek továbbra is lehetőségük van. A megosztott memóriát és szemafor a `rip` parancssal lehet letörölni.

2.4. A K2 rendszer újraindítása

Ha a PGW valamilyen ok miatt leáll (pl. véglegesen leszakad az MMTS rendszerről), akkor a teljes K2-t újra kell indítani. A PGW újraindítását a recovery logfile-ba írásával támogatja, ahova a felhasználói rendszer felől érkező ajánlatbeviteli kérések kerülnek. Az újraindítás során a pgw szintén létrehozza a szemafor és a megosztott memóriát, és feltölti azokat adattal, de az ajánlatbeviteli táblát a recovery log file-ből tölti fel. Ha a logfile is megsérült, akkor lehetőség van a következő ajánlatbeviteli rekord azonosítójának megadására.

Az IFSS processz csak a kapcsolatok felvételére szolgál, emlékezettel nem bír. Újraindítása megegyezik a normál indítással. Az IFSS kiszolgáló processzei az elkészült és feltöltött megosztott memóriára kapcsolódnak fel, és a felhasználó bejelentkezési adatát tárolják, minden egyes kapcsolat felvételekor új processz jön létre.

2.5. Az IFSC kapcsolat újraépítése (IFSC recovery)

Ha egy felhasználói rendszer kapcsolat (IFSC kapcsolat) leszakad, akkor a felhasználói rendszernek kell újra kezdeményezni a kapcsolat felvételt. A kapcsolat újbóli kialakulása után a felhasználói rendszer a lekérdezések során minden adatot újból megkap, azonban a következő módszerrel lehetősége van a leszakadás előtti lekérdezések kihagyására:

1. Kapcsolat felvétel (`ifsc_connect` függvény hívása) esetén a felhasználói rendszer megkapja a K2 indítási azonosítót. Ha ez az azonosító megegyezik a korábbi kapcsolatkor megkapottal, akkor ugyanaz a PGW fut ugyanazon megosztott memórián. Ekkor lehetőség van a leszakadás előtti lekérdezéskor megkapott változásszámok beállítására:
2. A felhasználói rendszernek lehetősége van bármely tábla lekérdezésekor használt változásszám beállítására az IFSC memóriájában az `ifsc_set_get_seq` függvény segítségével. Az adott táblához tartozó aktuális változásszámot a korábbi lekérdezés során kapta meg a lekérdező. A következő lekérdezés során (`ifsc_get_next_record`) az IFSC a megadott változásszámmal kérdezi le a következő rekordot az adott táblára vonatkozóan.

FIGYELEM! Ha a K2 rendszert újraindították, annak belső indítási azonosítója megváltozik, így a kliens oldalon a „rég” (újraindítás előtti) változásszámok használata *adatvesztést* eredményezhet (a kliens nem kap meg minden adatot), ezért javasoljuk, hogy a kliens oldalt erre készítsék fel és ha megváltozik a K2 belső indítási azonosítója, akkor a kliens oldali változásszámokat nullázni kell.

3. Futási paraméterek beállítása

A futási paraméterek beállítása a PGW-nél és az IFSS-nél is általában három szinten lehetséges. Ezek az alábbiak.

1. A program által meghatározott alapértelmezett (default) értékek szintje (befordítva található a programban)
2. A konfigurációs fájlban szereplő értékek szintje
3. Az indítási parancssorban megadott értékek szintje

Az egyes szinteknél a magasabb sorszám jelenti a nagyobb prioritást. Ez azt jelenti, hogy a program az indulás után a paramétereinek először alapértelmezett értéket ad, majd felolvassa a konfigurációs file-ját és az abban talált értékekkel felülírja az alapértelmezetteket. Ezután a parancssor feldolgozás következik³, amely szintén felülírja a megadott értékeket. A program csak ezután kezdi ellátni tényleges funkcióit.

3.1. A PGW futtatásához szükséges környezeti változók

A PGW futtatásához a megfelelő környezeti változók beállítása telepítéskor megtörtént.

A K2 rendszertelepítés után setup.sh és az aststools.sh scriptek felvételre kerülnek a rendszer indításához is használt k2 user .profile állományába. Ha ezt elvégezték, akkor utána egyéb kézi beállításra nincs szükség. Az aststools.sh az aststools komponensek és általában az MMTS környezet (TSMR, SYSSRV, AMP) eléréséhez szükséges path és környezeti változó beállításokat tartalmazza. A setup.sh ellenőrzi az aststools.sh által elvégzendő beállítások meglétét, illetve belépéskor még a programindítások előtt elvégzi a pgw és az ifss számára szükséges path és környezeti változó beállításokat.

3.2. A PGW futási paramétereinek beállítása

3.2.1. Alapértelmezett paraméterek

A PGW futási paraméterek alapértelmezett értékeit maga a program forráskód tartalmazza.

3.2.2. Konfigurációs file

A konfigurációs file-t (a helyét és nevét) a megfelelő környezeti változóban keresi a program. Ha a környezeti változó nem létezik, akkor egy pgw.cfg file-t próbál megnyitni az indítási könyvtárban.

3.2.3. Parancssor paraméterek

A PGW parancssori paraméterei az alábbiak lehetnek:

³ A parancssorban csak bizonyos paraméterek adhatóak meg

- b**<**maxbadreplymsg**> A maximálisan eltárolható ajánlatbeviteli hibaüzenetek száma (a *badreplymsg* tábla hossza.) A *maxbadreplymsg* paramétert írja felül.
- f** A PGW az előtérben fog futni. A *foreground* paramétert írja felül.
- F**<**firstorderentry**> A PGW azonosítóval látja el az ajánlatbevitelt (beszámozza azokat). A számozás a paraméterrel megadott számmal fog kezdődni. A *firstorderentry* paramétert írja felül.
- m**<**maxorderentry**> A maximálisan bevihető ajánlatok száma (az *orderentry* tábla hossza). A *maxorderentry* paramétert írja felül.
- N**<**numoforderbook**> A figyelésre felvehető ajánlatkönyvek maximális száma. A kétféle listában külön-külön. A *numoforderbook* paramétert írja felül.
- S** A PGW nem hajtja végre a restart logfile ellenőrzést. (De fel fogja olvasni!) A *skiplogchk* paramétert írja felül.
- p**<**password**> Az MMTS felhasználó jelszava. Ha nincs megadva, akkor indulás után a PGW bekéri. A *password* paramétert írja felül.
- P** A PGW ebben az esetben egyszerű jelszóváltó eszközként viselkedik. Bekéri a felhasználó jelszavát, bejelentkezik az MMTS-be, új jelszót kér a felhasználótól, majd megismétli a kérést ellenőrzés céljából. Ezután az MMTS-ben módosítja a felhasználó jelszavát és kijelentkezik onnan. Az **-u**<**username**> kapcsoló segítségével a kívánt felhasználónév megadható, a kapcsoló hiánya esetén a konfigurációs file-ban megadott felhasználó jelszava lesz megváltoztatva. A PGW *ebben az esetben semmilyen adatlekérdezést nem hajt végre*. A *chgpwd* paramétert írja felül.
- r**<**orderratio**> Az MMTS és a PGW ajánlati táblájának aránya. A PGW megkapja az MMTS-től az ajánlati táblájának hosszát és ezt elosztva az **orderratio**-val kapja meg saját táblájának hosszát. E mögött az áll, hogy a PGW csak saját (cégének) ajánlatait kapja meg és ez lényegesen kevesebb az MMTS-be bevihető ajánlatok számánál. A *orderratio* paramétert írja felül.
- t**<**seconds**> A PGW lekérdezési ciklusideje másodpercekben. Itt csak egész értéket lehet megadni. A *cycletime* paramétert írja felül.
- u**<**username**> A PGW ezzel a felhasználónévvel fog bejelentkezni az MMTS-be. A *username* paramétert írja felül.
- v** A PGW az alapértelmezett kimenetére írja az MMTS-től vett adatokat. Minden adatot kiír, csak valamilyen vizsgálat céljából érdemes használni. A *verbose* paramétert írja felül.

3.3. Az IFSS futtatásához szükséges környezeti változók

Az IFSS futtatásához a szükséges környezeti változó beállítások telepítéskor megtörténtek.

A K2 rendszertelepítés után *setup.sh* és az *aststools.sh* scriptek felvételre kerülnek a rendszer indításához is használt *k2 user .profile* állományába. Ha ezt elvégezték, akkor utána egyéb kézi beállításra nincs szükség. Az *aststools.sh* az *aststools* komponensek és általában az MMTS környezet (TSMR, SYSSRV, AMP) eléréséhez szükséges path és környezeti változó beállításokat tartalmazza. A *setup.sh* ellenőrzi az *aststools.sh* által elvégzendő beállítások meglétét, illetve belépéskor még a programindítások előtt elvégzi a *pgw* és az *ifss* számára szükséges path és környezeti változó beállításokat.

3.4. Az IFSS futási paramétereinek beállítása

3.4.1. Alapértelmezett paraméterek

A paraméterek alapértelmezett értékeit maga a program forráskód tartalmazza.

3.4.2. Konfigurációs file

A konfigurációs file-t (helyét és nevét) a megfelelő környezeti változóban keresi a program. Ha a környezeti változó nem létezik, akkor egy `pgw.cfg` file-t próbál megnyitni az indítási könyvtárban.

3.4.3. Parancssor paraméterek

-f Az IFSS előtérben fut. A `foreground` paramétert írja felül.

-s <service> Service név, ahol az IFSS várja a csatlakozásokat. A `ifssservice` paramétert írja felül.

3.5. Az IFSS felhasználók és jogosultságok nyilvántartása

IFS user A felhasználói rendszer, amikor felveszi a kapcsolatot a K2/IFSS-sel, akkor megad egy IFS felhasználó nevet és jelszót. Az IFSS ennek alapján kezeli a további kéréseket. Minden felhasználónévhez jogosultságok tartoznak. Ezek a felhasználóneveket csak az IFSS használja, az MMTS rendszerbe nem továbbítja.

Az IFSS felhasználók különböző jogokkal rendelkezhetnek, amelyet az IFSS lekérdezéskor ellenőriz, azaz a felhasználó csak olyan üzenetekkel operálhat a K2-n, amelyekhez joga van. A felhasználói jogok adatbázisa az `IFSS.ua` file-ban található. A beállítható jogok az alábbiak:

query Adatok lekérdezéséhez való jog.

entry Ajánlatok bevitele, módosítása és törlése.

confirm Bevitt ajánlatok megerősítése a K2-ben.

config Ajánlati könyv lista konfigurálása.

bypass A bevitt ajánlatok automatikusan megerősítettek lesznek.

admin Egyenlőre nem használt.

A user authorisation file (`uaf`) az IFSS felhasználói számlák adatait tartalmazza. Példa beállítások megtalálhatók a `cfg/IFSS.ua` file-ban. Egy sor egy felhasználó adatait tartalmazza kettősponttal (`,:`) elválasztva:

username IFSS felhasználó neve.

password A felhasználó jelszava.

status A felhasználó állapota, lehet aktív „a”, vagy felfüggesztett „s”.

privileges A jogosultságok vesszővel vannak egymástól elválasztva.

Minta .uaf felhasználói adatok file tartalmára:

```
#
user1:password:a:entry,confirm,query,config,admin,bypass
user2:password:a:query
```

Megjegyzés: a Távkereskedő az általa megvásárolt K2 gépen a IFSS.uaf állományt szabadon megváltoztathatja.

3.6. A K2 rendszer konfigurálhatósága

3.6.1. A K2 telepítések felvitt konfigurációs fájl

A K2 rendszer telepítések a BÉT által felírt konfigurációs fájlok kezdeti tartalmának megváltoztatása nem javasolt. Az azonnali és a derivatív rendszerhez csatlakozó K2a és K2d konfigurációs fájljainak szerkezete azonos.

3.6.2. A K2 felhasználó által nem módosítható konfigurációs paraméterek

A konfigurációs fájlok bizonyos paramétereinek értékét a K2 felhasználó azért sem módosíthatja, mert azok a licenc fájl által védettek. Ez azt jelenti, hogy amennyiben a felhasználó a fenti paraméterekre a telepítéskor megadott értékeket megváltoztatja, akkor a K2 program nem indul el.

3.6.3. A BÉT által használt LINUX/UNIX felhasználói azonosító és annak jogosultságai

A telepítéskor a BÉT munkatársai beállítanak egy felhasználót a K2 szerveren (Linux vagy Unix). Ennek jelszavát csak a BÉT munkatársai ismerik. Erre a felhasználóra a szoftver frissítések elvégzése miatt, a monitorozás és ellenőrzés miatt van szükség. A Távkereskedőnek a szerződés értelmében nem szabad korlátoznia a BÉT által használt felhasználó jogosultságait, nem szabad jelszavát módosítania.

4. A K2 rendszer működésének nyomon követése - Hibakeresés

4.1. A PGW és az IFSS processzek megfigyelhetősége

A PGW két log file-t ír:

Esemény log Hiba és nyomkövetési üzeneteket írja a pgw1.log, illetve pgw2.log nevű fájlokba. Az üzenetek magyarázata megtalálható az A Függelékben.

Restart log A file neve ÉÉÉÉHHHNN_1.rlg, illetve ÉÉÉÉHHHNN_2.rlg formátumú, ahol ÉÉÉÉ az évszám, HHH a hónap neve, NN nap. Ide az ajánlatbeviteli adatok kerülnek beírásra, nevezetesen ajánlatbevitel és ajánlatbeviteli státuszváltozás. PGW újraindítása során ezt a file-t olvassa fel az ajánlatbeviteli rekordok visszaállításához.

Mindkét log file helye a konfigurációs file-ban⁵ megadható (log_path és rlog_path bejegyzés). Az esemény log szöveges file, a szokásos módon olvasható. A restart log tartalmának értelmezésére egy rldump1, illetve rldump2 nevű eszköz áll rendelkezésre (lásd 10).

A PGW process állapotáról az IFSC library használata esetén a megosztott memória lastquerytime és pgwstate mezői adnak tájékoztatást, amit az ifsc_get_systime függvénnyel lehet lekérdezni. A PGW process minden lekérdezési ciklus végén a lastquerytime mezőbe beírja a pillanatnyi időt. Ez által megtudhatjuk, hogy mikor dolgozta fel a legutolsó lekérdezés és ajánlatbevitel válaszait. A watchpgwenbled konfigurációs paraméter megadása esetén a statikus adatok letöltése után a PGW process kettévál, a gyerek process folytatja a lekérdezéseket és ajánlati kérések beküldését, az anya process pedig csak wait rendszereljárás segítségével a gyerek process állapotát figyeli. Ha a gyerek process kilépett, akkor a PgwState mezőbe beállítja a PGWTERMINATED állapotot. (A kettéválásig PGWINITED állapotban van, és utána a kilépésig PGWRUNNING állapotban.)

Az IFSS által írt log a külön eseménylog:

Eseménylog Hiba és nyomkövetési üzeneteket írja az ifss1.log, ill. ifss2.log nevű file-okba. Az üzenetek magyarázata megtalálható az A Függelékben.

Rlg file-ok Recovery céljára a PGW-nél már említett log-ok szolgálnak. A file neve ÉÉÉÉHHHNN_1.rlg, illetve ÉÉÉÉHHHNN_2.rlg formátumú, ahol ÉÉÉÉ az évszám, HHH a hónap neve, NN nap. Ide az ajánlatbeviteli adatok kerülnek beírásra. PGW újraindítása során ezt a file-t olvassa fel az ajánlatbeviteli rekordok visszaállításához.

4.2. A recovery log fájl tartalma

rldump1 <filename> vagy rldump2 <filename> parancs segítségével a <filename> újraindítás log file tartalmát lehet ASCII formátumban kiírni a normál kimenetre. Az újraindítási log file az ajánlatbeviteli kéréseket és az ajánlatbeviteli kérések állapotváltoztatásait tartalmazza időbélyeggel ellátva. Formátuma a következő¹:

Táblázat C.1: Ajánlatbeviteli rekord – MMTS1

| Név | Hossz |
|----------------|-------|
| Idx | 6 |
| Timestamp | 20 |
| orderid | 6 |
| Time | 6 |
| OrdNo | 6 |
| TrdAccId | 12 |
| BuySell | 1 |
| OrderMethodSet | 32 |
| BoardId | 4 |
| SecId | 12 |
| Price | 10 |
| Yield | 10 |

⁵ Erre mutat a PGW1CONFIG vagy PGW2CONFIG.

¹ Az egyes mezők közt szeparátorok vannak: Idx és timestamp közt -, timestamp után - ... , többi közt |

| | |
|--------------|----|
| Quantity | 10 |
| Hidden | 10 |
| BrokerRef | 40 |
| ExpDate | 8 |
| ExpTime | 6 |
| TriggerPrice | 10 |
| SettleDate | 8 |
| MinFillQty | 10 |
| OUserIdx | 6 |
| FirmIdx | 6 |
| OrderDate | 8 |
| Status | 1 |
| Type | 1 |
| MsgIdx | 6 |

Táblázat C.2: Ajánlatbeviteli rekord – MMTS2

| Név | Hossz |
|---------------------|-------|
| orderid | 4 |
| time | 4 |
| OrdNo | 22 |
| OrdNoSpeedIdx | 4 |
| TrdAccId | 13 |
| BuySell | 1 |
| OrderType | 1 |
| Duration | 1 |
| PurgeOnLogoff | 1 |
| AllowSoftQtyLimit | 1 |
| AllowSoftPriceLimit | 1 |
| PositionType | 1 |
| IsPrivate | 1 |
| BoardId | 5 |
| SecId | 21 |
| Price | 12 |
| Yield | 12 |
| Quantity | 4 |
| VisibleQuantity | 4 |
| BrokerRef | 41 |
| ExpTime | 8 |
| TriggerPrice | 12 |
| TradeRef | 31 |
| MinFillQty | 4 |
| OpCode | 1 |
| PopCode | 1 |
| MultiLegOrdNo | 21 |
| MultiLegSpeedIdx | 4 |
| InstrId | 5 |
| UserId | 13 |
| FirmId | 13 |
| OrderDate | 8 |
| MarketMaker | 1 |
| Status | 1 |
| Type | 1 |
| MsgIdx | 4 |

Táblázat C.3: Ajánlatbeviteli állapot váltás

| Név | Hossz |
|-----------|-------|
| Idx | 6 |
| Timestamp | 20 |
| Status | 1 |

Táblázat C.4: Ajánlatbeviteli elutasított állapot

| Név | Hossz |
|-----------|-------|
| Idx | 6 |
| Timestamp | 20 |
| Status | 1 |
| MsgIdx | 6 |
| Msg max | 255 |

Példa:

```
000000-1999-Sep-20 11:01:01-->005001|110101|000000|0013-0000001|...
000000-1999-Sep-20 11:01:02-->U|
000000-1999-Sep-20 11:01:02-->E|
```

Az `rldump` figyelmeztető üzenetet ad, ha a file tartalma nem hasonlít egy restart logfile-éhez, illetve abban az esetben is ha `rldump1`-gyel 2-es logfile-t olvastunk fel vagy fordítva.

4.3. A megosztott memória tartalmának kiíratása

A `dump1` vagy `dump2` parancs a megosztott memória tartalmát írja ki táblánként karakteres formátumban. A megosztott memória azonosítóját a `PGW1CONFIG`, illetve `PGW2CONFIG` környezeti változóban megadott file-ból olvassa fel. Indulásakor felkapcsolódik a megosztott memóriára és szabad szemaforra vár. Majd a szemafor segítségével levédi a megosztott memóriát és kiírja az adatokat a normál kimenetre. Ezután a szemafort visszaállítja.

4.4. A K2 rendszer állapotának monitorozása

Rendszer állapot figyelő - MEM

A `mem1 [-fcn][-s<service|port>][-a<host|network>][-t<time>]` vagy `mem2 [-fcn][-s<service|port>][-a<host|network>][-t<time>]` parancs segítségével egy figyelő processzt lehet elindítani, ami a megosztott memória olvasásával a rendszer állapotáról gyűjt információt és UDP segítségével elküldi a megadott hálózaton vagy host-ra.

Indítási kapcsolók

A parancs kapcsolói:

-f A PGW az előtérben fog futni.

- c** Folytonosan fut. A kapcsoló nélkül csak egyszer gyűjt információt és utána kilép.
- n** A kigyűjtött információkat UDP segítségével elküldi. A kapcsoló nélkül csak a normál kimenetre írja ki.
- t<seconds>** A lekérdezési ciklusideje másodpercekben.
- s<service|port>** A megadott port-ra küldi a kigyűjtött információt. Ha service név van megadva, akkor a futtató számítógép segítségével feloldja a port számmá.
- a<host|network>** A megadott host-ra vagy network-re küldi el a kigyűjtött információkat. Ha host név van megadva, akkor a futtató számítógép segítségével feloldja a host nevet címé.

Beállítások

A MEM futtatásához szükség van a következők beállítására:

PGW1CONFIG környezeti változó A K2_a PGW konfigurációs file-ra kell hivatkoznia.

PGW2CONFIG környezeti változó A K2_d PGW konfigurációs file-ra kell hivatkoznia.

PGW konfigurációs file A megosztott memória és szemafor azonosítóját kell tartalmaznia. Lásd 3.3.2.

service Az operációs rendszer service adatbázisába a **service** bejegyzést fel kell venni.

host Az operációs rendszer host adatbázisába a **host** bejegyzést fel kell venni.

Működés

Indulása után a megosztott memóriára felkapcsolódik és szabad szemaforra vár. Majd ciklikusan a megosztott memória olvasása által a következő információkat gyűjti ki és küldi el:

- megosztott memória és szemafor azonosítók
- memória táblák méretei, telítettségei
- stats info a TE-ből: statsversion, logtime, transaction num, rejection num
- PGW, IFSS és MEM processzek információi: pid, ppid, ucomm, vshmsize, shmsize, start, utime, stime, process állapot

A MEM kimenete:

```
==> yellow 1
KEY: 0x70677731 SHMID: 420 SEMID: 570
-----Num
Max Recsize
market 3 3 48
sector 22 22 44
instr 11 11 48
board 9 9 88
secboard 260 260 752
firm 422 422 312
user 345 346 328
order 34 10000 248
trade 0 60000 224
negdeal 0 1000 172
orderentry 0 1000 188
badreplymsg 0 500 256
-----Last
orderno: 76
```

```

Last tradeno: 0
Last last_n tradeno: 0
Last negdealno: 76
Stats Version: 0x37f89a80
Log Time: 37839
Transactions: 33543
Rejections: 1
-----pid
ppid comm vshm shm starttime utime stime status
00003121 00000001 ifss 0 0 939047573 0 0 Sleep
00003113 00000001 pgw 4294 4294 939047561 0 0 Sleep
00003125 00000001 mem 4294 4294 939047585 0 0 Run
=====

```

Az első sorban (==>-val kezdődik) a gép nevét, és az üzenet sorszámát találjuk. Ha hálózaton továbbítjuk az üzeneteket, akkor a vevő ebből a sorszámból tudja megállapítani, hogy volt-e kimaradt üzenet¹.

4.5. A monitorozás eredményének rögzítése

ear.pl – MEM figyelő és értelmező

Az ear.pl egy rövid Perl script, amely a MEM által hálózaton kibocsátott csomagokat figyeli. A vett csomagokat host-onként külön file-ba írja. Figyeli a táblák telítettségét és erről hibaüzenet ír az ear.log hiba file-ba. Akkor is hibaüzenet kerül a file-ba, ha a PGW vagy IFSS processzek bármelyike nem fut. Szintén hibaüzenet az eredménye annak, ha egy host-ról régóta nem jött csomag (feltételezve, hogy több K2 üzeme esetén azok azonos gyakorisággal küldik a csomagokat). A fenti üzenetekről a -m parancssori opció beállítása esetén automatikusan E-mail-t küld a megadott címre.

Az ear.pl az alábbi kapcsolókkal vezérelhető:

- p<port> A port száma, amelyen az ear.pl figyeli a beérkező csomagokat.
- f Megadása esetén az ear.pl az előtérben fog futni.
- w<warning level> A figyelmeztetési szint százalékban, tizedes alakban. (pl.: 0,80 azt jelenti, hogy a táblák 80%-os betelítésénél lesz figyelmeztető üzenet.)
- e<error level> A hibaszint százalékban az előző pontban írtak szerint.
- t<timeout> A csomagvételi timeout.
- m<E-mail address> E-mail cím.
- h A fentiek kiírása a standard kimenetre.

4.6. A PGW által használt erőforrások megjelenítése

Erőforrás megmutató szkript – sipc [1|2] A sipc parancs a PGW által használt erőforrásokat mutatja meg.

¹ A szállító UDP protokoll üzenetet veszthet

4.7. Erőforrás felszabadítás a PGW vagy IFSS hibás kilépése után

erőforrás kitakarító szkript – `ripcc` [1|2] A PGW leállása után megmaradó rendszer erőforrásokat (megosztott memóriát és szemafor) törli. A PGW és IFSS process kilépése során a rendszer erőforrásokat (megosztott memóriát és szemafor) nem törli le, így lehetőség van az esetleges hibás kilépés vizsgálatára. A `ripcc` parancs ezeket az erőforrásokat távolítja el, amire a következő indítás előtt szükség van.

4.8. K2 felhasználói jelszó módosítása

jelszó változtató szkript – `k2changepwd` [1|2] A K2_a vagy K2_d PGW processzét indítja el -P kapcsolóval az MMTS jelszó megváltoztatásához. Az éles MMTS kereskedési rendszerbe történő első belépéskor kötelező a jelszó megváltoztatása. Eerre szolgálhat a jelszó változtató szkript.

5. A K2 és az IFSC együttműködését bemutató demo és teszt programok üzemeltetése

Az IFSC használatának bemutatására és tesztelésére a következő demo programok készültek:

get_table Tábla lekérdezés IFSC segítségével.

get_ob Ajánlati könyv lekérdezés IFSC segítségével.

ob_watch Ajánlati könyv figyelés beállítása IFSC segítségével.

send_order Egy ajánlati kérés elküldése IFSC segítségével.

gen_order Ajánlati kérések készítése ajánlati könyv figyelésével.

demo Rövid példa program tábla lekérdezés, ajánlat bevitel, ajánlatbeviteli rekord lekérdezés és IFSC segédfüggvények együttes használatára.

5.1. `get_table`

A `gettable` <-1|-2> <table>[-ctqig][-h<host>][-s<service>][-u<user>][-p<password>] parancssal elindított program az IFSC könyvtár segítségével a megadott tábla tartalmát kérdezi le.

Indítási kapcsolók

A parancs kapcsolói:

-c Folytonosan fut, a következő ciklusban csak a változásokat kérdezi le.

-t A nyomkövetési mód.

-q Változásszám mód (alapértelmezett).

-i Index mód.

-g Mezőváltozás mód.

-h<host> Az IFSS ezen a host-on található.

-s<service> Az IFSS ezzel megadott porton várja a felkapcsolódni kívánó klienseket.

-u<user> Ifss felhasználói név, amivel a kliens bejelentkezik.

-p<password> Ifss felhasználó jelszava.

-1 K2_a szerverhez kapcsolódik. Ha a -2 is meg van adva, akkor mindkét szerverhez kapcsolódik.

-2 K2_d szerverhez kapcsolódik. Ha a -1 is meg van adva, akkor mindkét szerverhez kapcsolódik. A K2_d szerverhez tartozó service értéket csak az IFS2SERVICE környezeti változóval lehet megadni.

<table> Lekérdezett tábla neve, vagy all, amely utóbbi segítségével minden táblát lekérdez.

A host, service, user és password paraméterek az IFSHOST, IFSSERVICE, IFSUSER és IFSPWD környezeti változókkal is megadhatóak.

Beállítások

A get_table futtatásához szükség van a következők beállítására:

service Az operációs rendszer service adatbázisába (/etc/services) a service bejegyzést fel kell venni.

host Az operációs rendszer host adatbázisába (/etc/hosts) a host bejegyzést fel kell venni.

Működés

Indulása után az ifsc_connect függvény segítségével bejelentkezik az IFSS kiszolgálóba, majd ciklikusan meghívja az ifsc_get_next_record függvényt és a megkapott adatokat karakteres formátummá alakítva kiírja a normál kimenetre.

5.2. get_ob

A get_ob <secboardid>[-cmt][-h<host>][-s<service>][-u<user>]
[-p<password>] paranccsal elindított program az IFSC könyvtár segítségével a megadott értékpapírtábla ajánlati könyvét kérdezi le.

Indítási kapcsolók

A parancs kapcsolói:

<secboardid> Az értékpapírtábla azonosítója.

-c Folytonosan fut, mindem másodpercben a teljes ajánlati könyvet lekérdezi.

-m Összesített ajánlatikönyv lekérdezés. (A kibontott az alapértelmezett)

-t A nyomkövetési mód.

A többi kapcsoló leírását lásd a get_table-nél

Csak a -s<service> beállítással lehet megadni, hogy a K2_a vagy K2_d szerverhez kapcsolódjon.

Az indítás előtti beállításokat lásd a get_table-nél.

Működése

Indulása után az `ifsc_connect` függvény segítségével bejelentkezik az IFSS kiszolgálóba, majd ciklikusan meghívja az `ifsc_get_next_orderbook` függvényt és a megkapott adatokat karakteres formátummá alakítva kiírja a normál kimenetre.

5.3. *ob_watch*

Az `ob_watch <secboardid>|-l[-rmt] [-h<host>][-s<service>] [-u<user>][-p<password>]` paranccsal elindított program az IFSC könyvtár segítségével a megadott értékpapírtábla ajánlati könyv figyelését állítja be.

Indítási kapcsolók

A parancs kapcsolói:

<secboardid> Az értékpapírtábla azonosítója.

-l Beállítás helyett lekérdezi, mely értékpapírokra van figyelés beállítva.

-r Hozzáadás helyett törli az ajánlati könyv figyelést az adott értékpapírra vonatkozóan.

-m Összesített ajánlatikönyv lekérdezés. (A kibontott az alapértelmezett)

-t A nyomkövetési mód.

A többi kapcsoló leírását lásd a `get_table`-nél

Csak a `-s<service>` beállítással lehet megadni, hogy a K2_a vagy K2_d szerverhez kapcsolódjon.

Az indítás előtti beállításokat lásd a `get_table`-nél.

Működése

Indulása után az `ifsc_connect` függvény segítségével bejelentkezik az IFSS kiszolgálóba, majd meghívja az `ifsc_orderbook_conf` függvényt, vagy ha lekérdezés kértek, akkor az `ifsc_get_orderbook_list` függvényt és a megkapott adatokat karakteres formátummá alakítva kiírja a normál kimenetre.

5.4. *send_order*

A `send_order <-1|-2> -e<E|A|W>[-f<orders>][-t][-h<host>] [-s<service>][-u<user>][-p<password>]` paranccsal elindított program az IFSC könyvtár segítségével ajánlatbeviteli kéréseket küld a kiszolgálónak.

Indítási kapcsolók

A parancs kapcsolói:

-e<E|A|W> Ajánlatbeviteli kérések típusa: E (entries) újak beadása, A (amends) módosítások, W (withdrawns) törlések

-f<orders> File neve, ahol az ajánlatbeviteli rekordok találhatóak. Ha nincs megadva, akkor a normál bemenetről veszi azokat.

-t A nyomkövetési mód.

A többi kapcsoló leírását lásd a `get_table`-nél

Az indítás előtti beállításokat lásd ugyancsak a `get_table`-nél.

Ajánlatbeviteli rekord formátuma

A beolvasott ajánlatbeviteli rekordok mezői megegyeznek az ifsc_orderentry által elfogadott mezőkkel. A formátumuk karakteres, elválasztó karakter a '|'. Egy ajánlatbeviteli rekord egy sor.

Működése

Indulása után az ifsc_connect függvény segítségével bejelentkezik az IFSS kiszolgálóba, majd a felolvasott ajánlatbeviteli rekordot IFSC formátumba átalakítva meghívja az ifsc_orderentry függvényt és kiírja a bevitt rekordok ajánlatszámát.

5.5. gen_order

A gen_order <boardid> <secid> <B|S> <price> <qty> <trdacc> [-ct][-h<host>][-s<service>][-u<user>][-p<password>] paranccsal elindított program az IFSC könyvtár segítségével ajánlatbeviteli kéréseket állít össze. Az adott értékpapír ajánlati könyvét figyelve az adott árnál jobb ajánlatokra kötés képes ajánlatokat állít össze a megadott mennyiség erejéig. Az összeállított ajánlati kéréseket send_order segítségével el lehet küldeni a kiszolgáló számára. Az így kialakított program egy automatát képez, amely a piacon lévő ajánlatokat figyelve kereskedik.

Indítási kapcsolók

A parancs kapcsolói:

<boardid> Értékpapír tábla azonosító.

<secid> Értékpapír azonosító.

<B|S> Vétel vagy eladás.

<price> Ár, amely vagy annál jobb ajánlatokra reagál.

<qty> Maximális mennyiség, amely erejéig a program az ajánlatokat összeállítja.

<trdacc> Kereskedési számla azonosító.

-c Folytonos működés. Ha nincs megadva, akkor csak egy ajánlati könyv lekérdezés segítségével állítja össze az ajánlatokat.

-t A nyomkövetési mód.

a többi kapcsoló leírását lásd a get_table-nél

Jelenleg csak a K2_a rendszerrel működik együtt. Az indítás előtti beállításokat lásd a get_table-nél.

Működése

Indulása után az ifsc_connect függvény segítségével bejelentkezik az IFSS kiszolgálóba, majd ciklikusan az ifsc_orderbook_conf függvény segítségével lekérdezi az aktuális ajánlati könyvet, annak elemzésével ajánlatbeviteli rekordokat állít össze és a normál kimenetre kiírja azokat.

5.6. Demo

A demo [-h<host>][-s<service>][-u<user>][-p<password>] paranccsal elindított program az IFSC könyvtár segítségével a megadott tábla tartalmát kérdezi le.

Indítási kapcsolók

A parancs kapcsolói:

-h<host> Az IFSS ezen a host-on található.

-s<service> Az IFSS ezzel megadott porton várja a felkapcsolódni kívánó klienseket.

-u<user> Ifss felhasználói név, amivel a kliens bejelentkezik.

-p<password> IFSS felhasználó jelszava.

A `host`, `service`, `user` és `password` megadhatók az `IFSHOST`, `IFSSERVICE`, `IFSUSER` és `IFSPWD` környezeti változókkal is megadhatóak. Jelenleg csak a K2_a rendszerrel működik.

Beállítások

A demo futtatásához szükség van a következők beállítására:

service Az operációs rendszer service adatbázisába a `service` bejegyzést fel kell venni.

host Az operációs rendszer host adatbázisába a `host` bejegyzést fel kell venni.

Működése

Indulása után az `ifsc_connect` függvény segítségével bejelentkezik az IFSS kiszolgálóba, majd ciklikusan, amíg adatot kap, az `ifsc_get_next_record` függvény segítségével lekérdezi az értékpapír rekordokat és a megkapott adatokat karakteres formátummá alakítva, kiírja a normál kimenetre. Ezek után összeállít egy ajánlatbevittelt és elküldi a K2-be, majd ismét az `ifsc_get_next_record` függvény segítségével lekérdezi az ajánlatbevitteli rekordokat, ahol az utolsó ajánlatbevitteli rekord az előtte bevitt rekord. Végül `ifsc_disconnect` függvénnyel lebontja a kapcsolatot.

5.7. Adatformátum átalakító awk programok

get_ob.awk Ajánlati könyv figyelésének eredményét táblázat formátumban jeleníti meg.

get_table.awk Tábla lekérdezések eredményét táblázat formátumban jeleníti meg.

pobo.awk Kereskedési munkaállomás saját ajánlati könyv file-t átalakítja K2 ajánlatbevitteli kérés formátumává.

5.7.1. get_ob.awk

A bemenetén beolvasott sorokat ajánlati könyvnek megfelelően értelmezi (egy sor egy teljes ajánlati könyvnek felel meg) és azt táblázat formátumot alkotó karakteres sorokat ír ki a normál kimenetre. Az ajánlati könyv adatokat a `get_ob` program képes előállítani.

5.7.2. get_table.awk

A bemenetén beolvasott sorokat tábla lekérdezés egy rekordjának értelmezi és a tábla típusának megfelelően sorrendezi, válogatja ki és tabulálja a mezőket és a normál kimenetre kiírja. A tábla lekérdezési rekordokat a `get_table` programmal lehet előállítani.

5.7.3. pobo.awk

A bemenetén beolvasott sorokat kereskedési állomás saját ajánlati könyv formátumnak megfelelően értelmezi, a referenciákat feloldva és a mezők sorrendjét átcserélve IFSC ajánlatbevitteli formátummá alakítja. A referencia feloldáshoz az indulás során `get_table` `secboard` és `get_table board` parancsokkal két referencia tömböt állít össze.

5.8. ifstest: Felhasználói programok tesztelése

A felhasználói alkalmazás, próbaprogram `libifstest.sl/libifstest.so/ifstest.dll` könyvtárral történő linkelése esetén lehetőség van az IFSC függvényhívások K2 kapcsolati kiszolgáló nélkül történő tesztelésére, az alkalmazói programozói interface kipróbálására, az alkalmazott adatstruktúrák használatának ellenőrzésére.

A `libifstest.sl/libifstest.so/ifstest.dll` ugyanazokat az IFSC függvényeket tartalmazza (Unix, Linux és Windows NT) rendszerekre és hasonlóan működik, mint a `libifsc.sl/libifsc.so/ifsc.dll` könyvtárak. A `create` és `connect` hívások kiadása azonban nem épít ki TCP/IP socket kapcsolatot, hanem a visszaadott kapcsolati leíró a tesztelt program indítási könyvtárában megnyitott `ifscdata.dat` file-ra mutat és az IFSC kéréseket abból szolgálja ki. A függvényhívások által visszaadott táblák, mezők értékei tehát ilyenkor nem az MMTS-hez csatlakozó K2 rendszerből, hanem ebből az adatfájlból származnak. A teszteléshez ezért az indítási könyvtárban léteznie kell az `ifstest` könyvtárral együtt adott teszt adatfájlnak.

A `libifstest.sl/libifstest.so/ifstest.dll` csak csökkentett kéréseket tud kiszolgáltatni, azaz nem alkalmas az `ifsc` függvényhívások teljes körének tesztelésére és a tetszőleges paraméterekkel végzett próbákra. A mellékelt `apitest.c` demo program a kiszolgálható kérésekre ad mintát, és a forráskód elemzése alapján a teszt program vagy próba alkalmazás keretei is kialakíthatók. A K2 nélküli tesztelésnél tehát a próbaalkalmazásban azokat az `ifsc_` hívásokat érdemes tesztelni és olyan paraméter használatával, amelyek az `apitest.c` forrásprogramjában megtalálhatók. Az `apitest.c` ezen kívül az `ifscdata.dat` teszt adatállomány felvételére is alkalmas, amennyiben `-c` opcióval indítva működő K2 rendszerhez csatlakoztatjuk. Erre a teszt állomány felvételre azonban nincs szükség, mivel azt a BÉT-en előre megtették és az `ifstest`-hez adott `ifscdata1.dat` valamint `ifscdata2.dat` fájlokban a teszt adatok megtalálhatók.

A $K2_a$ azonnali és $K2_d$ derivatív kapcsolatok tesztelésére ugyanis külön teszt adat file készült: az `ifscdata1.dat` valamint `ifscdata2.dat` az `mmts1`-nek és az `mmts2`-nek megfelelően. Egyszerre csak az egyikkel tesztelhetünk, úgy, hogy a megfelelő file-t a próbált program indítási könyvtárában lévő `ifscdata.dat` nevű file-ba másoljuk le.

6. Függelék A: Hibaüzenetek

A PGW és IFSS processzek és az IFSC programozói könyvtár a következő hiba- és nyomkövetési üzeneteket adhatnak¹.

BADREPLY - Bad reply received when sent 0x%x

Az MMTS rendszer `MSGTYPE_REPLY_BAD` választ küldött a PGW lekérdezésére. A lekérdező üzenet száma hexadecimális alakban van.

BADREPMSGFULL - Not enough space for new badreplymsg (buffer size is %d)!

Ajánlatbeviteli hibaüzenetek számára nincs több hely.

BINDFAILED - %s - unable to bind address!

A megadott figyelési címtartomány megadása sikertelen.

BRDCHGINVLEN - board %s change message length, supposed %d, received %d

¹ ABC sorrendben, azonosítójuk szerint

Az MMTS által a MSGTYPE_BOARD_CHANGE_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

BRDINV - board %s doesn't exists

Az MMTS által a MSGTYPE_BOARD_CHANGE_REQ lekérdezésre adott válaszban szereplő board ismeretlen.

BRDINVLEN - board %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_BOARD_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

CANNOTFORK - cannot fork!

Új process létrehozási hiba.

CHECKSPACEERR - check_space() failed

A szabad hely ellenőrzés hibát jelzett - új adat számára nincs szabad hely a megosztott memóriában.

CLIENTLICEXCEED - The number of client %d is greather then in the license.

A licencben megadott kliensek száma meghaladásra került.

COMPILED - Compiled: %s %s

Információ a program verziójáról, fordítási idejéről stb..

CONNACCEPETD - %s: Accepting a connection from %s port %u

A kiszolgáló a következő helyről elfogadta a felkapcsolódást.

CONNECTFAILED - IFS_COMERROR - %s - unable connect to %s!

Nem tudott az adott kiszolgálóhoz kapcsolatot létrehozni.

CONNECTTRACE - Connected to %s on port %u at %s

IFSC a következő kiszolgálóra csatlakozott.

DISCONNECT - disconnecting

IFSC lekapcsolódik.

DOWNLOADSUCC - All data has been downloaded.

Tájékoztató üzenet az első lekérdezési periódus végén. Jelzi, hogy a statikus adatok és a változások sikeresen lekérdezésre kerültek.

EOFRECEIVED - Receiving end of file without disconnect message - connection with %s aborted!

A kapcsolat az IFSC-vel megszakadt.

FOPENERR - %s - %s

File megnyitási hiba.

FORCELOGGEDOFF - The MMTS user of pgw force logged off

A pgw MMTS felhasználóját kiléptették.

FRMCHGINVLEN - firm %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_FIRM_CHANGE_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

FRMINV - firm %s doesn't exists

Az MMTS által a MSGTYPE_FIRM_CHANGE_REQ lekérdezésre adott válaszban szereplő firm ismeretlen.

FRMINVLEN - firm %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_FIRM_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

INSCHGINVLEN - instr %s change message length, supposed %d, received %d

Az MMTS által a MSGTYPE_INSTR_CHANGE_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

INSINV - instr %s doesn't exists

Az MMTS által a MSGTYPE_INSTR_CHANGE_REQ lekérdezésre adott válaszban szereplő instrumentum ismeretlen.

INSINVLEN - instr %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_INSTR_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

INVALIDFIRM - The %s user's firm doesn't match the firm given in the config file

Az MMTS felhasználó cége más, mint ami a konfigurációs file-ban megadtak.

INVALIDLICENSE - The license given in the config file is invalid.

A licenc bejegyzés a konfigurációs file-ban érvénytelen.

INVLINE - %s – invalid line

A konfigurációs file sora nem értelmezhető.

INVOBCTION - Unknown watch action in watch config ask of SecBoard %s

A megadott ajánlatfigyelési mód nem érvényes.

INVPRIVILEGE - User discarded, invalid privilege %s in record %d

A megadott IFSS felhasználói jogosultság érvénytelen érték.

INVSTATUS - Invalid status (%c) received!

A megadott Ajánlatbeviteli rekord kért állapota érvénytelen.

INVSTATUSCHNG - Status (%c) of orderentry %d can not be changed!

A megadott ajánlatbeviteli rekord állapota nem megváltoztatható.

INVTRANSCODE - Unknown transition code %d!

A megadott Ajánlatbeviteli kérés típusa érvénytelen.

INVUSERNAME - User discarded, name %s in record %d

A megadott IFSS felhasználói név túl hosszú, vagy üres.

KILLFAILED - Kill failed - %s

Hiba szignál küldés (kill hívás) során.

LINETOOLONG - %s – line too long

A konfigurációs file sora túl hosszú.

LISTEN_FAILED - %s - unable to listen on socket!

A kiszolgáló nem tud figyelni a létrehozott socket-en.

LNTRDINVLEN - last_n_trades no %d message length, supposed %d, received %d

Az MMTS által a MSGTYPE_LAST_N_TRADES_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

LOGAPPEND - Log file (%s) opened for append.

Sikeres log file megnyitás hozzáfűzésre.

LOGONFAILED - Logon failed

A bejelentkezés az MMTS rendszerbe nem sikerült.

LOGONINVREPLY - Logon failed, invalid reply from host

A bejelentkezés az MMTS rendszerbe nem sikerült - az MMTS válasza nem értelmezhető.

LOGONSUCCESS - Logon successful

A bejelentkezés az MMTS rendszerbe sikeresen megtörtént.

LOGOPENERR - Log file (%s) cannot be opened, so stderr will be used.

Log file megnyitási hiba. A hibaüzenetek a továbbiakban a *stderr*-re íródnak.

MBPADDED - Market by price of SecBoard %s added to watch.

A megadott értékpapír összesített ajánlati könyve figyelésre került.

MBPLISTED - Market by price of SecBoard %s has been already watching!

A megadott értékpapír összesített ajánlati könyve már szerepel a figyelési listában.

MBPLISTFULL - Not free place to watch market by price of SecBoard %s.

Új összesített ajánlati könyv figyelésére nincs szabad hely - egy másik értékpapír ajánlati könyv figyelését törölni kell.

MBPNOTFOUND - Market by price of SecBoard %s can not found!

A megadott értékpapír összesített ajánlati könyve nem figyelt - nem lehet a figyelését kikapcsolni.

MBPNOTLISTED - Market by price of SecBoard %s can not found!

Összesített ajánlati könyv az adott értékpapírhoz nem található - az ajánlati könyv figyelését előbb be kell állítani.

MBPQUERY - %s market by price query; seq %d.

IFSC a következő összesített ajánlati könyv adatot vette.

MBPREMOVED - Market by price of SecBoard %s removed from watch.

A megadott értékpapír összesített ajánlati könyve a továbbiakban nem kerül figyelésre.

MEMHOSTNAMEERR - gethostname() failed

A *gethostname()* függvény nem tudta megadni a host nevét.

MEMNONET - Network initialisation failed, stdout will be used

A hálózat inicializálása sikertelen, a kimenet a továbbiakban az *stdout* lesz.

MEMSENDTOERR - sendto() error - %s

Az üzenetküldés a socket-re sikertelen.

MEMWRITEERR - write() error - %s

A *stdout*-ra történő írás sikertelen.

MKTCHGINVLEN -market %s change message length, supposed %d, received %d

Az MMTS által a MSGTYPE_MARKET_CHANGE_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

MKTINV - market %s doesn't exists

Az MMTS által a MSGTYPE_MARKET_CHANGE_REQ lekérdezésre adott válaszban szereplő market ismeretlen.

MKTINVLEN - market %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_MARKET_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

MSGERROR - IFS_MSGERROR - The received len %d is differ from len %d in header!

A vett üzenet hossza nem azonos a fejlécben megadottal.

MSGPROTVERDIFF - The IFS_PROTOCOLVERSION %d of ifss differs from version %d of ifsc.

A szerver IFS_MSG_PROTOCOL_VERSION-ja különbözik a kliens könyvtár verziójától.

MSGTOOLONG - IFS_MSGERROR - The length of message %d is greater then length of buffer %d!

Az üzenet túl hosszú.

NEGDINVLEN - negdeal no %d message length, supposed %d, received %d

Fix ügyeletek lekérdezésre az MMTS nem megfelelő hosszú üzenettel válaszolt.

NOCONFIGPRIV - no config privilege %s

Az IFSC felhasználónak nincs ajánlati könyv beállítási joga.

NOCONFIRMPRIV - no confirm privilege %c

Az IFSC felhasználónak nincs megerősítési joga.

NOENTRYPRIV - no entry privilege %d.

Az IFSC felhasználónak nincs beviteli joga.

NOHOSTBYNAME - IFS_COMERROR - gethostbyname() failed, host: <%s> !

A *gethostbyname()* hibával tért vissza - a kiszolgáló neve nem feloldható.

NOLICENSE - The %s query has not been licensed.

A kért funkcióra nincs licence.

NOPGWCONF - PGWCONFIG doesn't exist or too long

A PGWCONFIG környezeti változó nincs beállítva, vagy az értéke túl hosszú.

NOQUERYPRIV - no query privilege %d.

Az IFSC felhasználónak nincs lekérdezési jogosultsága.

NOSECBOARD - SecBoard %s can not found!

A megadott értékpapír nem található.

NOSERVBYNAME - IFS_COMERROR - getservbyname() failed, service: <%s> !

A *getservbyname()* hibával tért vissza - a kiszolgáló port neve nem feloldható.

NOSOCKNAME - IFS_COMERROR - %s - unable to read socket address!

IFSC a kiszolgáló címét nem tudta lekérdezni.

NOSUCHMSG - Unhandled message %d !

Nem kiszolgálható üzenet.

NOSUCHTABLE - Unhandled table %d.

A megadott táblalekérdezés nem megválaszolható.

NOTENOUGHMEM - malloc() failed at stats

Nincs elegendő memória.

OBADDED - OrderBook of SecBoard %s added to watch

A megadott értékpapír kibontott ajánlatikönyve figyelésre került.

OBINVLEN - byorder message length, supposed %d, received %d

Az MMTS által a MSGTYPE_MARKET_BYORDER_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

OBINVSECB – by order secboard %s doesn't exists

Az MMTS által a MSGTYPE_MARKET_BYORDER_REQ lekérdezésre adott válaszban szereplő secboard ismeretlen.

OBLISTED - OrderBook of SecBoard %s has been already watching!

A megadott értékpapír kibontott ajánlati könyve már szerepel a figyelési listában.

OBLISTFULL - Not free place to watch OrderBook of SecBoard %s

Új kibontott ajánlati könyv figyelésére nincs szabad hely - egy másik értékpapír ajánlati könyv figyelését törölni kell.

OBNOTFOUND - OrderBook of SecBoard %s can not found!

A megadott értékpapír kibontott ajánlati könyve nem figyelt - nem lehet a figyelését kikapcsolni.

OBNOTLISTED - OrderBook of SecBoard %s can not found!

Kibontott ajánlati könyv az adott értékpapírhoz nem található - az ajánlati könyv figyelmet előbb be kell állítani.

OBQUERY - %s orderbook query; seq %d.

IFSC a következő kibontott ajánlati könyv adatot vette.

OBREMOVED - OrderBook of SecBoard %s removed from watch

A megadott értékpapír kibontott ajánlati könyve a továbbiakban nem kerül figyelésre.

OEADDED - Orderentry added (orderid: %d trans_code %d)

Ajánlatbeviteli kérés sikeresen hozzáadva.

OEINVREPLY - orderentry status remains unknown idx: %d

Az MMTS által a MSGTYPE_ORDER_REQ üzenetre adott válasz értelmetlen.

OENOTCSTRING - IFS_OENOTCSTRING - The orderentry is not NULL terminated!

Az ajánlatbeviteli kérés nem NULL lezárású C string.

OENOTEXISTS - Orderentry (orderid %d) can not found!

A megadott ajánlatbeviteli rekord nem található.

OESTATUSCHNG - Status %c of Orderentry (orderid %d) changed to %c

A megadott ajánlatbeviteli rekord állapota sikeresen megváltoztatásra került.

OETABLEFULL - Not enough space for new orderentry (buffer size is %d)!

Ajánlatbeviteli kérdések számára nincs több hely.

OETOOLONG - IFS_OETOOLONG - The order_len %d is greather then len of the aviable buffer!

Az ajánlatbeviteli rekord túl hosszú.

OETOOMANYMSG - more than two message received at orderentry

Az MMTS által a MSGTYPE_ORDER_REQ üzenetre adott válasz több mint két üzenetet tartalmaz. Az első üzenet az ajánlatbevitel sorsára vonatkozik, a második a szokásos záró null-üzenet.

OETYPEINV - Invalid orderentry type %c idx: %d

Hibás ajánlatbevitel típus. Az ajánlatbevitel típus lehet:

- bevitel
- módosítás
- törlés.

OPTARGREQ - Option -%c requires an argument

A parancssorban megadott kapcsolóhoz kell, hogy tartozzon érték.

OPTUNKNOWN - Unrecognized option: - %c

Ismeretlen parancssori kapcsoló.

ORDINVLEN - order no %d message length, supposed %d, received %d

Az MMTS által a MSGTYPE_ORDER_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

PARAMDECIMAL - %s : %d

Decimális konfigurációs paraméter.

PARAMDOUBLE - %s : %e

Double konfigurációs paraméter.

PARAMHEXNUM - %s : 0x%x

Hexadecimális konfigurációs paraméter.

PARAMSTRING - %s : %s

String konfigurációs paraméter.

PGWTERMINATED - The pgw process terminated!

A Pgw process leállt.

PWDCHGATTEMPT - MMTS password change attempt.

Kezdeményezés az MMTS jelszó megváltoztatására.

PWDCHGFAILURE - MMTS password change failure %s.

MMTS jelszó megváltoztatása nem sikerült.

PWDCHGSUCCESS - MMTS password changed successfully.

MMTS jelszó sikeresen megváltoztatva.

RECONOK - reconnect succeeded.

Újrakapcsolódás sikeresen megtörtént.

RECONSTART - reconnect starting.

Újrakapcsolódás elindult.

RECRESMBP - receive repsonse of market by price; status: %d

Az IFSC a következő összesített ajánlati könyv adatokat vette.

RECRESMBPCONF - receive repsonse of market by price conf; status: %d

Az IFSC a következő összesített ajánlati könyv beállítási választ kapta.

RECRESMBPLIST - receive repsonse of market by price list; status: %d

Az IFSC a következő összesített ajánlati könyv lista adatokat vette.

RECRESPOB - receive repsonse of orderbook; status: %d

Az IFSC a következő kibontott ajánlati könyv adatokat vette.

RECRESPOBCONF - receive repsonse of orderbook conf; status: %d

Az IFSC a következő kibontott ajánlati könyv beállítási választ kapta.

RECRESPOBLIST - receive repsonse of orderbook list; status: %d

Az IFSC a következő kibontott ajánlati könyv lista adatokat vette.

RECRESPOE - receive repsonse of orderentry; status: %d

Az IFSC a következő ajánlatbeviteli választ kapta.

RECRESPTAB - receive repsonse of %d; status: %d

Az IFSC a következő választ vette.

RECVFAILED - IFS_COMERROR - %s - Error at recv!

Hiba az üzenet vétele során.

RECVTRACE - recv >%s<

Az IFSC a következő üzenetet vette.

REPLYBAD - Bad reply received!

A pgw process kérdésre BadReply (hiba) üzenetet kapott.

RESTRICTEDLOGON - The pgw logged on with RESTRECTED mode - You must change your MMTS password!

A pgw csak korlátozott módban tudott bejelentkezni az MMTS rendszerben - Jelszót kell változtatni!

RLDFILEOPENERR - The restart log file (%s) cannot be opened - %s

Az *rldump* nem tudja megnyitni restart log file-t.

RLDINVFILE - This file (%s) doesn't seem to be a restart log file!

Az *rldump* szerint a file nem restart log file.

RLDINVRECTYPE - Invalid record type %d in record %d

A rekord típus ismeretlen az *rldump* számára.

RLOGDATAREAD - Couldn't read the data from the restart log file - %s

A restart file-ből történő adat olvasás sikertelen.

RLOGDATAWRT - Couldn't write the data into the restart log file - %s

A restart file-ba történő adat írás sikertelen.

RLOGHEADREAD - Couldn't read the header (index/type/length) from the restart log file - %s

A restart file-ből történő fejléc olvasás sikertelen.

RLOGHEADWRT - Couldn't write the header (index/type/length) into the restart log file - %s

A restart file-ből történő fejléc írás sikertelen.

RLOGINVIDX - Invalid index in the restart log file - recno: %d

Hibás index a restart log file-ban. Ilyen ajánlatbeviteli rekord index nem létezik.

RLOGINVMSGIDX - Invalid message index in the restart log file - recno: %d

Érvénytelen hibáüzenet index a restart log file-ban.

RLOGINVRECTYP - Invalid record type read from the restart log - record: %d type: %d

Nemlétező ajánlatbeviteli rekord típus. Az alábbi típusok léteznek:

- ajánlatbevitel az IFSS-től
- ajánlatbevitel státusz változás
- MMTS az ajánlatbevittet visszautasította.

RLOGINVTIMESTAMP - The existing restart log file timestamp is different from the shared memory timestamp

A restart log file-ban lévő időbélyeg nem egyezik meg az MMTS időbélyegével.

RLOGINVRLOGFILE - The MMTS type of existing restart log file is different from the file reader.

A restart log file MMTS típusa (I. v. II.) különbözik a file beolvasóétól (program MMTS típusától.).

RLOGNAMEFAILED - Restart log file name cannot be created

Újraindítási file-t nevét nem lehet létrehozni.

RLOGNOEND - Cannot go to the end of the restart log file - %s

Nem lehetséges az újraindítási file végére pozicionálni.

RLOGOPENFAILED - Restart log file name cannot be opened

Az újraindítási file-t nem lehet megnyitni.

RLOGREPORT - %d record read from the restart log - number of orderentries %d

Tájékoztató üzenet arról, hogy hány rekord olvasása történt meg az újraindítási file-ból.

RLOGREWDERR - Couldn't rewind the restart log file - %s

Nem lehetséges az újraindítási file elejére pozicionálni.

RLOGTIMEREADERR - Couldn't read the timestamp from the restart log file - %s

Az időbélyeg nem olvasható fel az újraindítási file-ból.

RLOGTIMEWRERR - Timestamp cannot be written to the restart log file - %s

Az időbélyeg nem írható bele az újraindítási file-ba.

RLOGTYPEADERR - Couldn't read the system type from the restart log file - %s

Az MMTS típus nem írható bele az újraindítási file-ba.

RLOGTIMEWRERR - System type cannot be written to the restart log file - %s

Az MMTS típus nem írható bele az újraindítási file-ba.

SBDIELDNOERR - secboard %s number of fields, counted %d, received %d

Az MMTS-től kapott MSGTYPE_SECBOARD_CHANGE_REP üzenetben a mező ok tényleges száma nem egyezik meg az üzenetben közölt mező számmal.

SBDINV - secboard %s doesn't exists

Az MMTS-től kapott MSGTYPE_SECBOARD_CHANGE_REP üzenetben a secboard ismeretlen.

SBDINFIELD - secboard %s invalid field indicator %c

Az MMTS-től kapott MSGTYPE_SECBOARD_CHANGE_REP üzenetben a mező indicator érvénytelen.

SBDINLEN - secboard %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_SECBOARD_CHANGE_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

SCTINLEN - sector %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_SECTOR_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

SEMCREATEFAILED - %s - create_sem() failed

A szemafort nem sikerült létrehozni.

SEMDELETEFAILED - %s - delete_sem() failed

A szemafort nem sikerült törölni.

SEMGETFAILED - %s - get_sem() failed

A szemafor értékét nem sikerült lekérdezni.

SEMOPENFAILED - %s - open_sem() failed

A szemafort nem sikerült megnyitni.

SEMSETFAILED - %s - set_sem() failed

A szemafort nem sikerült beállítani.

SEMWAITFAILED %s - wait_sem() failed

A szemaforra várakozás hibával tért vissza.

SENDFAILED - IFS_COMERROR - %s - Error at send (sent byte: %d)

Hiba az üzenet elküldése során.

SENDTRACE - send >%s<

IFS a következő üzenetet küldte el.

SEQHASHUPDERR - Cannot update seqhash, index: %d num: %d

A változásszám referenciát nem sikerült kitölteni.

SETSOCKOPTFAILED - %s - Error at setsockopt: REUSEADDR

REUSEADDR socket paramétert nem sikerült beállítani.

SHMATFAILED - %s - shmat() failed

A megosztott memóriára nem sikerült felkapcsolódni.

SHMCREATEFAILED - %s - create shmget() failed

A megosztott memória nem hozható létre.

SHMDETACH - Detached from shared memory

Sikeres lekapcsolódás a megosztott memóriáról.

SHMDTFAILED - %s - shmdt() failed

A megosztott memóriáról nem sikerült lekapcsolódni.

SHMINFOFAILED - %s - shmctl() -info- failed

A megosztott memóriáról információ nem kérdezhető le.

SHMMEMSIZE - Shared memory size: %d bytes

A megosztott memória mérete.

SHMOPENFAILED - %s - open shmget() failed

A megosztott memória nem nyitható meg.

SHMREMOVEFAILED - %s - shmget() -remove- failed

A megosztott memóriát nem sikerült törölni.

SHMWAIT - wait for shared memory, key: 0x%x

Megosztott memória elérésre várakozik.

SOCKETFAILED - IFS_COMERROR - %s - unable to create socket!

Socket létrehozási hiba.

STARTSUCCESS - Successfully started

A PGW sikeresen elindult. A háttérben fut, és létrehozta a szemafort. Ezután fog majd bejelentkezni az MMTS-be.

STATSINVLEN - stats message length, supposed %d, received %d

Az MMTS által a MSGTYPE_STATS_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

SYSTIMEINVLEN - systime message length, supposed %d, received %d

Hiba a Rendszeridő üzenet hosszánál.

SYSTIMESYNCE - offset is %d seconds

A rendszeridő eltolódás.

SYSTIMESYNCF - query time is %e seconds

Lekérdezési idő.

TRACEFILEERR - IFS_TRACEFILEERR - trace file (%s) cannot be opened.

IFSC nyomkövetési file (ifstrace.log) megnyitási hiba.

TRACESET - trace set to %d

IFSC nyomkövetési mód a következő értékre lett beállítva.

TRDINVLEN - trade no %d message length, supposed %d, received %d

Az MMTS által a MSGTYPE_TRADE_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

TRDLOST - last_n_trades no %d is an old trade but cannot be found (last trdno %d)

Az MMTS által a MSGTYPE_LAST_N_TRADE_REQ lekérdezésre adott válaszban szereplő kötés nem található.

TSMRDISCONNECT - Disconnected from the TE

A TSMR üzenet kapcsolati réteg lekapcsolódott a kereskedési rendszerről.

TSMRERRMSG - %s

TSMR üzenet kapcsolati réteg hibát észlelt.

TSMRLOGOFF - Logged off from TE

A kereskedési rendszerből kijelentkezett.

UAFFULL - There is no room for record %d in UAF file (max %d)

IFSS felhasználói adatok számára nincs több szabad rekord.

UNKNOWNMSG - Unknown TSMR message 0x%x

Ismeretlen TSMR üzenet kapcsolati réteg hiba.

USERINTERR - %s internal error.

Program belső hiba.

USERINVPWD - %s password invalid.

Az hibás IFSC felhasználói jelszó.

USERLOGGEDIN - %s logged in.

A megadott IFSC felhasználó bejelentkezett.

USERLOGOUT - logout.

Az IFSC felhasználó kijelentkezett.

USERNOFIRM - %s user doesn't have firm

A felhasználóhoz nincs cég megadva.

USERNOTACTIVE - %s isn't active.

Az IFSC felhasználó inaktív.

USERNOTEXISTS - %s doesn't exists.

IFSC felhasználói azonosító nem létezik.

USERNOTLOGGEDIN - %s isn't logged in yet.

Az IFSC felhasználó nem jelentkezett még be.

USGDUMP - Usage: %s

dump segéd program parancssor leírás.

USGIFD - Usage: %s [-f] [-t<cycle>]

ifd (interface database) segéd program parancssor leírás.

USGIFSS - Usage: %s [-f]

IFSS parancssor leírás.

USGMEM - Usage: %s [-c] [-n] [-f] [-t<cycle>] [-s<service/port>] [-a<address/hostname>]

mem segéd program parancssor leírás.

USGPGW - Usage: %s [-f] [-v] [-S] [-t<cycle>] [-u<username>] [-p<password>]

PGW parancssor leírás.

USGUNKNOWN - Unknown program (%s), no usage available.

A programról nincs parancssor leírás.

USRCHGINVLEN - user %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_USER_CHANGE_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

USRIDNOTUNIQ - Userid isn't unique, %s in record %d

A megadott IFSS felhasználói név nem egyedi.

USRINV - user %s doesn't exists

Az MMTS által a MSGTYPE_USER_CHANGE_REQ lekérdezésre adott válaszban szereplő felhasználó nem létezik.

USRINVLEN - user %s message length, supposed %d, received %d

Az MMTS által a MSGTYPE_USER_REQ lekérdezésre adott válasz hossza hibás. Az üzenet feldolgozása ennek ellenére megtörtént.

WRONGPWD - User discarded, password %s in record %d

A megadott IFSS felhasználói jelszó túl hosszú vagy üres.

WRONGSTATUS - User discarded, invalid status %c in record %d

A megadott IFSS felhasználó státusza érvénytelen érték.