

# Magiszter

Számítástechnikai szerkesztőség

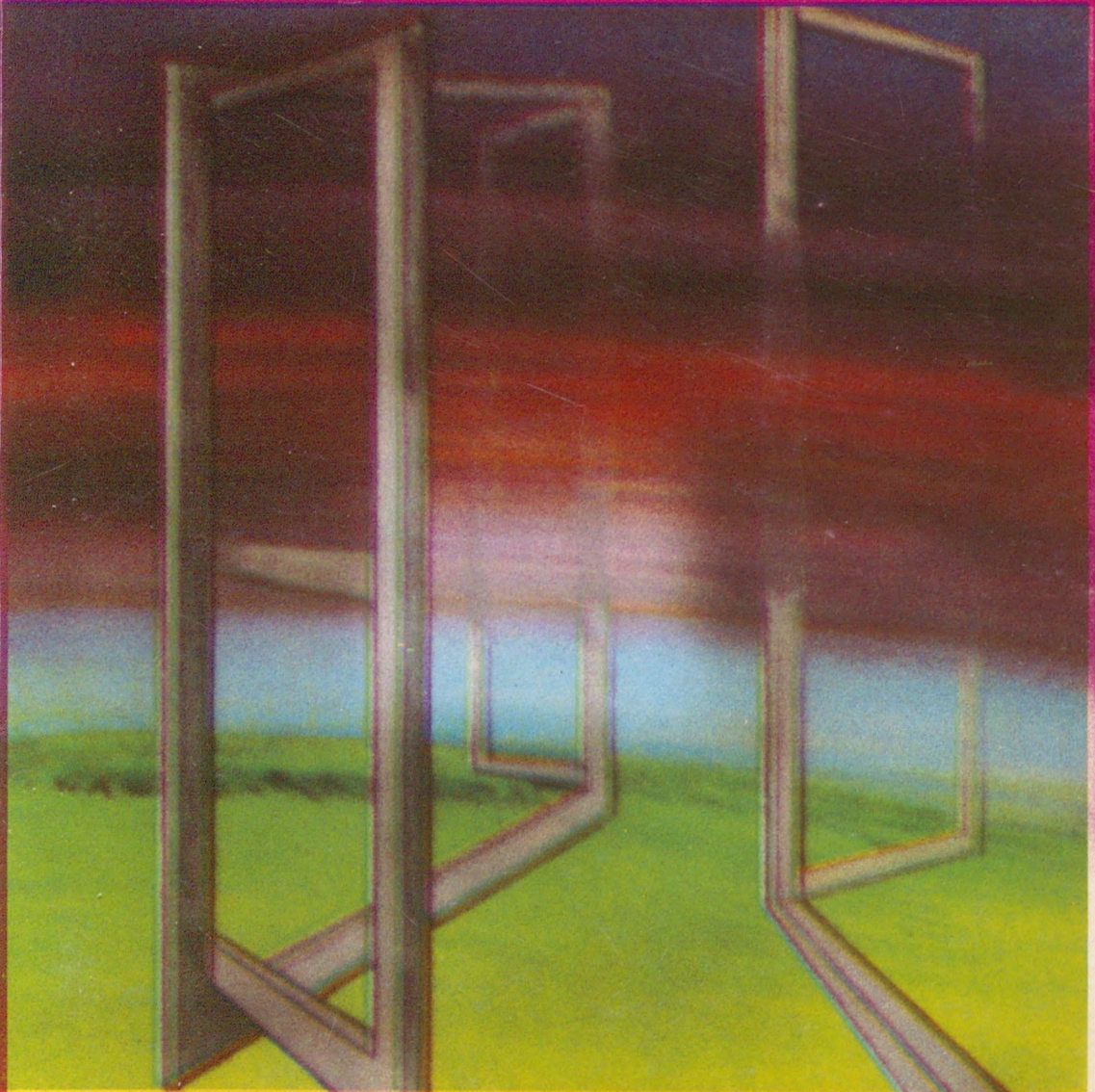
## CREAP

hierarchikus menükesztő és futtató rendszer

Programvédelem nélkül!

Szoftver  
IBM PC, XT, AT és  
kompatibilis mikroszámítógépekre

*Schmitt*



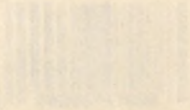


SCHILL RÓBERT

**CREAP**

Menü- és panelrendszer Generátor

MTA  
KIK



MTA KIK Budapest, 2008



193652

SCHILL RÓBERT

# CREAP

Hierarchikus Menü ill. Panelrendszer Generátor  
és  
Futtatórendszer

2.10 Változat

**MTA  
KIK**



**Akadémiai Kiadó, Budapest 1988**

885658

**Szoftver-dokumentáció**

**Sorozatszerkesztő: Pintér Tibor**

**A kiadványt szakmailag ellenőrizte: Juhász Lehel**

**© Akadémiai Kiadó és Nyomda Vállalat · Schill Róbert, 1988**

**Minden jog fenntartva**

MAGYAR  
AKADÉMIAI NYOMDA ÉS KIADÓ  
VÁLLALAT  
KÖNYVTÁRA

**A szoftver-dokumentáció a DOG dokumentáció generátorral készült**

**© Akadémiai Kiadó és Nyomda Vállalat · Schill—Szegedi, 1987**

M. TUD. AKADÉMIA KÖNYVTÁRA  
Könyvtár: 10901 /19... 88

## T a r t a l o m

1. Bevezetés.....	1.
2. A panelrendszer definiálása, karbantartása.....	4.
2.1. A CREAP rendszer indítása.....	6.
2.2. A CREAP által kezelt file-ok.....	7.
2.3. Kiszállás a rendszerből, output file-ok képzése	8.
2.4. A billentyűzet használata.....	9.
2.5. A paneltétel azonosítása a tétel nevével.....	13.
2.6. A paneltételek definiálása.....	15.
2.7. Új paneltétel beszúrása.....	16.
2.8. A paneltétel törlése.....	16.
2.9. Alpanelek kimentése és beolvasása.....	17.
2.10. Globális alpanelek.....	18.
2.11. Aktív panelek definiálása.....	19.
2.11.1. Kezdő C függvény meghatározása.....	20.
2.11.2. Befejező C függvény meghatározása.....	20.
2.12. Elszámoló rutinok beépítése.....	21.
2.13. Panelek átkötése, ismétlése.....	22.
2.14. Paneltételek sorrendjének megváltoztatása.....	25.
(felcserélés)	
2.15. Tétel típusának megváltoztatása.....	25.
2.16. Végpontok meghatározása.....	26.
2.17. C függvényt hívó végpont.....	26.
2.18. DOS parancssorozatot hívó végpont.....	28.
2.18.1. A szimbolikus paraméterek.....	29.
2.18.2. A kiterjesztett DOS parancsnyelv.....	31.
2.18.2.1. ASK.....	31.
2.18.2.2. BEEP.....	32.
2.18.2.3. EXIT.....	32.
2.18.2.4. GOTO.....	32.
2.18.2.5. IF.....	33.
2.18.2.6. IFN.....	34.
2.18.2.7. PAUSE.....	34.
2.18.2.8. PNLEXEC.....	35.
2.19. Az ablakok paramétereinek meghatározása.....	36.
2.19.1. Az ablakok helyének meghatározása.....	36.
2.19.2. Az ablakok méretének módosítása.....	37.
2.19.3. Az ablakok színeinek meghatározása.....	38.

2.19.4. A keret típusának állítása.....	39.
2.20. A tételek nevének módosítása.....	39.
2.21. Pull-down menük.....	40.
2.22. A paneltételeket azonosító karakterek.....	40.
meghatározása	
2.23. A helpfile illesztése a panelrendszerhez.....	41.
2.24. Kezelési segédlet megjelenítése.....	43.
2.25. A panelszerkezet megjelenítése és.....	43.
dokumentálása	
2.26. Kilépés a DOS-ba.....	44.
2.27. A CREAP verzió kiírása.....	44.
3. A futtatórendszer használata.....	45.
4. A felhasználói C függvények.....	48.
4.1. A végpontot lefedő C függvény.....	48.
4.2. Az aktív panel kezdőfüggvénye.....	49.
4.3. Az aktív panel befejezőfüggvénye.....	50.
4.4. Az elszámolást végző rutinok.....	51.
5. A felhasználó által használható függvények és.....	54.
változók	
5.1. A futtatórendszer panelkezelő függvényei.....	54.
5.1.1. inipanel().....	55.
5.1.2. dopanel().....	56.
5.1.3. PNL_protect().....	56.
5.1.4. PNL_isprotect().....	57.
5.1.5. PNL_glprotect().....	58.
5.1.6. PNL_glisprotect().....	59.
5.1.7. PNL_getarg().....	59.
5.1.8. PNL_setarg().....	60.
5.1.9. PNL_itnums().....	60.
5.1.10. PNL_name().....	61.
5.1.11. PNL_select().....	61.
5.1.12. PNL_exec().....	62.
5.2. A futtatórendszer általános függvényei.....	62.
5.2.1. getanswer().....	63.
5.2.2. message().....	63.
5.2.3. myerror().....	64.
5.2.4. myderror().....	64.
5.2.5. myalloc().....	65.

5.2.6. myrealloc().....	65.
5.2.7. convnum().....	66.
5.2.8. scanstr().....	67.
5.2.9. xchgstr().....	68.
5.2.10. newname().....	69.
5.3. A futtatórendszerbe beépített MSCEX függvények.	69.
5.4. A futtatórendszer változói.....	70.
6. A futtatórendszer operátori kezelése.....	71.
6.1. A felhasználói program indítása.....	71.
6.2. A felhasználói program szolgáltatásai.....	72.
7. Példa a CREAP futtatórendszer használatára.....	74.
7.1. A CENV program specifikációja.....	74.
7.2. A CENV panelrendszerének definiálása.....	75.
7.3. A CENV-ről a CREAP által készített dokumentáció	78.
7.4. A CENV.C keretprogram forrása.....	92.
7.5. A CENV.CP generált forrásfile.....	98.
7.6. A CENV.EXE generálása.....	99.
7.7. A CENV.EXE használata.....	99.
8. A disztributív lemez tartalma.....	101.



## 1. Bevezetés

Az alap- és alkalmazói szoftvert írók számára időigényes és fáradságos feladat az egyes szoftverek vezérlését lefedő menürendszerek megtervezése, programozása, dokumentálása és karbantartása. Ehhez nyújt segítséget a CREAP környezet, amely sok lehetőségével, rugalmasságával és kényelmes használatával ezt a munkát drasztikusan leegyszerűsíti.

Jelen leírásban a rendszerbe foglalt menüt panelnek nevezzük, tehát gyakorlatilag a panel és a menü kifejezések szinonimaként értelmezhetők.

A CREAP által generált panelrendszer végpontjai DOS parancssorozatok vagy C függvényhívások lehetnek, így a rendszer segítségével olyan hierarchikus panelrendszerek hozhatók létre, amelyek alkalmasak

- jól paraméterezzhető DOS parancssorozatok megfelelő döntési ágon történő elindításához és/vagy
- a programozó által írt C függvények hierarchikus rendszerbe-foglalásához.

Igy a CREAP rendszer segítségével definiálható, majd ezek alapján vezérelhető egy olyan monitor, amely az alkalmazói rendszer egyes funkcióit a DOS parancssorozatból összeállított, vagy C függvény hívásából álló "végpont" kiválasztásával végzi el.

A CREAP felhasználása tipikusan két szintre oszlik:

1. csak DOS parancssorozatokat tartalmazó hierarchikus rendszer létrehozása:  
ebben az esetben jelen leírás C nyelvvvel kapcsolatos információi nem szükségesek a rendszer használatához;
2. C nyelven írt professzionális monitor létrehozása (a CREAP elsősorban ilyen célból készült):  
ekkor a felhasználónak (elsősorban a 7. fejezetben található példa tanulmányozásával) alaposan el kell sajátí-

tania a rendszer használatát.

A CREAP környezet használatakor élesen el kell különíteni a következő két fázist:

- a panelrendszer definiálása, avagy a már létrehozott panelrendszer módosítása a CREAP programmal,
- a panelrendszer használata a CREAP futtatórendszerének segítségével.

A CREAP hardver- és szoftverkönyezete:

- IBM XT/AT, ill. azokkal kompatibilis számítógép,
- PCDOS/MSDOS 3.xx operációs rendszer,
- forrásnyelvi programmal való összeépítés (C függvények hívása) esetén a Microsoft C 4.00 vagy 5.00 környezet.

Mind a generáló, mind pedig a létrehozott rendszer 80x25-ös alfanumerikus módban üzemel.

A CREAP rendszer elemei:

Generatív fázishoz:

- a CREAP.EXE a panelrendszer definiáláshoz, generáláshoz és karbantartáshoz,

Futtatási fázishoz:

- a PANEL.EXE standardizált monitorprogram olyan panelrendszer használatához, ahol a rendszerhez nem tartoznak C nyelven megírt függvények, azaz csak DOS végpontot tartalmaznak (ebben az esetben a felhasználó mentesül a fordítási és szerkesztési tevékenységektől; a panelrendszer felhasználása mindössze a PANEL.EXE program megfelelően paraméterezett elindítását jelenti),
- C nyelvű rutinokkal való összeépítés esetén a futtatórend-

szert tartalmazó PANEL.LIB könyvtár a felhasználó programjának létrehozásához.

A CREAP.EXE program angol nyelvű, tehát üzenetei, segítő információi és hibajelzései angol nyelvűek. A futtatórendszert megvalósító PANEL.EXE programnak, illetve a PANEL.LIB könyvtárnak ezzel szemben magyar nyelvű változata is van (PANELM.EXE és PANELM.LIB néven), ezek futtatás közben magyar nyelvű üzenetekkel működnek.

A CREAP által generált panelrendszer minden esetben egy tömörített file-ban tárolódik, tehát a generálási fázis és a futtató fázis ugyanannak a file-nak az alapján működik. Ez magasszintű kezelhetőséget biztosít a karbantartáshoz és a módosításhoz. A létrehozott programtermék induláskor ebből a file-ból tölti fel a szükséges adatokat, tehát a panelekre vonatkozó (C nyelvi részeket nem érintő) információk később módosíthatók a felhasználó rendszerének változatlanul hagyásával.

A futtatórendszert tartalmazó könyvtárak jelenleg csak Small memóriamodellben léteznek. A további memóriamodellekre való kifejlesztésük folyamatban van.

A következő fejezet a generálási és karbantartási fázist (azaz a CREAP program használatát) tárgyalja.

A 3. fejezet a futtatórendszer használatát körvonalazza.

A 4. fejezet a felhasználói C függvények megírásának szabályait tartalmazza.

Az 5. fejezet a futtatórendszernek a felhasználó által hívható rutinjait és azok beépítésének szabályait írja le.

A 6. fejezet a futtatórendszer operátori kezelését részletezi.

A 7. fejezet egy viszonylag összetett példán magyarázza a CREAP futtatórendszer használatát.

A 8. fejezet a disztributív lemezen található file-okat írja le.

## 2. A panelrendszer definiálása, karbantartása

A CREAP környezet használatának első fázisa a panelrendszer megtervezése és létrehozása a CREAP program segítségével.

A generálási fázisban a későbbi futtatási rendszer minden szolgáltatása (értelmes korlátozásokkal) elérhető, tehát a generálási fázis egy kibővített futtatási fázisnak felel meg a végpontok végrehajtásának szimulálásával.

A CREAP által biztosított lehetőségek:

- ablakos panelek létrehozása, módosítása,
- panelrendszer végpont definiálása DOS parancssorozattal,
- panelrendszer végpont definiálása C függvénnyel,
- alpanelek kimentése és beolvasása,
- ablakos panel átváltása "lehúzható" (továbbiakban Pull-down) menüformába és vissza,
- globális paneltétel hozzárendelése az F5, F6, F7 és F8 funkcióbillentyűkhöz,
- aktív panelek definiálása (amelyek a vezérlésen kívül feldolgozást is végeznek),
- "elszámoló" rutinok beépítése a végpanelek hívása elé és mögé,
- egyes panelek átkötése különböző panelágakra (ismétlés - duplikálás),
- paneltétel sorrendjének megváltoztatása (felcserélés),
- ablakok méretének és helyének beállítása,
- paneltételt azonosító karakter definiálása,

- direkt ugrás egy bármely ágon lévő megnevezett paneltételre,
- az ablakba be nem férő tételek kezelése (tehát egy panelen tetszőleges számú tétel szerepelhet - scroll, lapozás),
- ablakok kerettípusának meghatározása,
- színek meghatározása (alapszín, kiemelt szín, szerkesztési (editálási) szín, üzenetek színe, helpkiírási szín),
- helpfile illesztése a panelrendszerhez,
- nyomtatható dokumentáció előállítása a panelrendszerről,
- kilépés a DOS operációs rendszerbe.

A DOS parancssorozatot tartalmazó paneltételek számára

- szimbolikus paraméterezésre van lehetőség, amely futtatási időben módosítható, kimenthető és betölthető;
- a DOS Batch parancsértelmezőhöz képest kibővített parancsnyelv áll rendelkezésre.

A CREAP bármilyen gazdag lehetőségeket is kínál a menet közbeni tervezésre, indítás előtt szükséges a felhasználói rendszer bizonyos szintű megtervezése. A feladatspecifikáción túl eldönthető kérdések:

- mely végpontokat kell C függvényhívással, és melyeket DOS parancssorozattal megoldani? (kell-e egyáltalán C függvényhívás?)
- mely végpontok használják a képernyőt, azaz mely végpontoknál kell majd gondoskodni a képernyőtartalom visszaállításáról?
- a C függvényeket hívó végpontok megoldhatók-e közös C függvénnyel a végpont sorszáma segítségével? (a funkciók elvégzése táblázatosítható?)

- mely panelek legyenek Pull-down szerkezetűek?
- melyek legyenek a globális (F5..F8 billentyűvel bármely panelből elérhető) panelrészek?
- van-e szükség "aktív" (feldolgozást is végző) panelekre?
- van-e szükség panelek átkötésére, duplikálására?

## 2.1. A CREAP rendszer indítása

A CREAP rendszer generálási fázisa mindössze a CREAP.EXE programot (a panelgenrátort) igényli.

A program indítása a következő paranccsal lehetséges:

```
>[path]CREAP [-b] [-i] [panelfile]
```

ahol 'path' a CREAP.EXE file-t tartalmazó könyvtár (amennyiben a CREAP a DOS Environment path által megtalálható - lásd a DOS SET parancsát -, a 'path' megadása szükségtelen),

'-b' a fekete-fehér üzemmódot jelöli,

'-i' egy látványos nyitóképet kiírása,

'panelfile' pedig a módosítandó panelfile neve típus nélkül.

Új panelrendszer definiálásakor a 'panelfile' paraméter nem adható meg.

## 2.2. A CREAP által kezelt file-ok

A CREAP a következő file-okat kezeli:

- panelfile ".PNL" típussal:

Tömörítve tartalmazza a panelrendszer összes információját.

Az input panelfile kiválasztása csak a CREAP indításakor lehetséges.

Az output panelfile a rendszerből való kiszálláskor választható ki akkor, ha a panelrendszerben változás történt. Az output panelfile kiválasztása egy ablakban megjelenő lista alapján lehetséges.

A CREAP Alt-W (Write) és Alt-R (Read) részpanelkezelő funkciói mindig csak alpanel típusú tételekre vonatkozhatnak (l. az "Alpanelek kimentése és beolvasása" c. pontban).

- helpfile ".HLP" típussal:

A panelrendszer tételeihez tartozó help (segítő információk) szövegeit tartalmazza, tetszőleges szövegszerkesztővel kezelhető formában.

A panelrendszerhez tartozó helpfile "A helpfile illesztése" műveletnél adható meg (ablakos kiválasztással). A helpfile mindig csak input file.

- generált C modul (C függvényt hívó végpanel esetén) ".CP" típussal:

A .CP modul a panelrendszer által használt C függvények táblázatát tartalmazza, C forrásnyelvi formában.

A C modul generálása a rendszerből való kiszálláskor történhet, abban az esetben, ha minden C függvényt hívó végpont definiálása megtörtént. Az output file neve szintén ablakos kiválasztással határozható meg. A generált C modul a felhasználó C függvényeinek és a futtatórendszernek a kapcsolatát teszi lehetővé. E modult a felhasználó programjának létrehozásakor le kell fordítani és a felhasználói programhoz (természetesen a futtatórendszer könyvtárával együtt) hozzá kell szerkeszteni. Legegyszerűbb megoldás: generált .CP file beszurása a felhasználói program forrásába egy "#include" utasítással.

A CREAP által kezelt file-ok kötelezően egy (nem feltétlenül az aktuális) könyvtárban vannak.

### 2.3. Kiszállás a rendszerből, output file-ok képzése

A főpanel elhagyása a CREAP-ból való kiszállást jelenti. Ekkor - ha a panelrendszerben változás történt vagy új a panelrendszer - a képernyő jobb felső sarkában automatikusan megjelenik a létező panelfile-ok listája. A felhasználó ezek közül választhat, vagy az első file-név előtt szereplő "New file" (új file) tételt jelölheti meg. Amennyiben a felhasználó nem akarja kimenteni a panelt, az ESC billentyű leütésével letilthatja a kiírást. "New file" esetén a rendszer lekérdezi a file nevét (típust nem szabad megadni) és létrehozza ezt a file-t. A panelfile hibamentes kiírását egy "Write OK" üzenet jelzi, amelyet az ESC billentyű leütésével kell nyugtázni.

Ha a panelrendszer C függvényeket hív, akkor kiszálláskor a rendszer a "Do You want generated C module? (y/n)" (Akar C modult generáltatni?) kérdést teszi fel. 'Y' (igen) válasz esetén a képernyőn megjelenik a ".CP" kiterjesztésű file-ok ablaka, amelyen kiválasztható a kívánt file, illetve "New file" választása esetén megadható az új file neve.

A fejlesztés során a felhasználó bármikor felfüggesztheti a panelrendszer definiálását, mivel a CREAP kiszálláskor automatikusan kéri a panelfile-ba való kimentést, amely panelfile a

következő definiálási, illetve karbantartási munkafolyamat inputja lehet.

#### 2.4. A billentyűzet használata

A billentyűzet kiemelt jelentősége miatt ebben a pontban a generálás alatt érvényben lévő billentyűk jelentésének általános és vázlatos leírása található.

ESC

elhagyás, kilépés minden egyes funkcionál;

ENTER

az aktuális (kiemelt színnel jelölt) tétel kiválasztása;

? (ua., mint az Alt-F1)

általános információk és kezelési segédlet megjelenítése.

#### Numerikus billentyűzet - kurzorvezérlők

<felfelé nyíl> (8)

felfelé következő tétel kiválasztása;

<lefelé nyíl> (2)

lefelé következő tétel kiválasztása;

<balra nyíl> (4)

a bal oldali tétel kiválasztása (csak Pull-down menü esetén);

<jobbra nyíl> (6)

a jobb oldali tétel kiválasztása (csak Pull-down menü esetén);

PgUp (9)

lapozás felfelé;

PgDn (3)

lapozás lefelé;

Home (7)

ugrás a legelső tételre;

End (1)

ugrás a legutolsó tételre;

INS (0)

új tétel beszúrása az aktuális elé;

Ctrl-A (append)

új tétel hozzáadása a tételsor végéhez;

DEL (.)

az aktuális tétel eltörlése;

Ctrl-<balra nyíl> (Ctrl-4)

az ablak mozgatása balra;

Ctrl-<jobbra nyíl> (Ctrl-6)

az ablak mozgatása jobbra;

Ctrl-<felfelé nyíl> (Ctrl-8) vagy Ctrl-Home vagy Ctrl-PgUp

az ablak mozgatása fölfelé;

Ctrl-<lefelé nyíl> (Ctrl-2) vagy Ctrl-End vagy Ctrl-PgDn

az ablak mozgatása lefelé;

Shift-<lefelé nyíl> (Shift-2)

az ablak magasságának növelése;

Shift-<felfelé nyíl> (Shift-8)

az ablak magasságának csökkentése;

Shift-<jobbra nyíl> (Shift-6)

az ablak szélességének növelése;

Shift-<balra nyíl> (Shift-4)

az ablak szélességének csökkentése.

## Funkcióbillentyűk

Ctrl\_F1:

a főmenühöz rendelt help megjelenítése;

F1

az aktuális tételhez tartozó help megjelenítése;

F2

csak futtatási fázisban (szimbolikus paraméterek módosítása);

F3

csak futtatási fázisban (szimbolikus paraméterek beolvasása);

F4

csak futtatási fázisban (szimbolikus paraméterek kiírása);

F5

globális paneltétel definiálás (futtatási fázisban végrehajtás);

F6

globális paneltétel definiálás (futtatási fázisban végrehajtás);

F7

globális paneltétel definiálás (futtatási fázisban végrehajtás);

F8

globális paneltétel definiálás (futtatási fázisban végrehajtás);

F9

kilépés a DOS operációs rendszerbe;

F10

bármely ágon lévő tétel kiválasztása a tétel nevének megadásával és direkt ugrás erre a tételre (l. még "A paneltétel azonosítása a tétel nevével" c. pontban).

### Alt billentyűk

Alt-F1 (ua., mint a ?)

általános információk és kezelési segédlet megjelenítése;

Alt-A (Actual)

az aktuális tétel nevének módosítása;

Alt-C (Colors)

a panel és a globális színek beállítása;

Alt-D (Display, Document)

a panelrendszer információinak megjelenítése képernyőn, majd opcionálisan nyomtatható dokumentáció készítése;

Alt-E (End function)

az aktuális panel befejező C függvényének meghatározása;

Alt-F (Frame)

a panel kerettípusának meghatározása;

Alt-H (Help indexing)

a helpfile illesztése a panelrendszerhez (indexelés);

Alt-L (Letters)

a tételazonosító karakterek meghatározása;

Alt-M (Mark)

az aktuális tétel megjelölése duplikáláshoz, átkötéshez;

Alt-N (Name)

a főpanel nevének módosítása;

Alt-P (Pull down)

ablakos panel átalakítása Pull-down menüvé és viszont;

Alt-R (Read)

alpanel betöltése file-ból az aktuális tételbe;

Alt-S (Start function)

az aktuális panel kezdő C függvényének meghatározása;

Alt-T (Type)

az aktuális tétel típusának megváltoztatása;

Alt-V (Version)

a CREAP verzió megjelenítése;

Alt-W (Write)

az aktuális tétel (alpanel) kimentése file-ba;

Alt-X (eXchange)

az aktuális tétel felcserélése egy másik tétellel (sorrendváltás).

### Egyéb billentyűk

Ha a leütött billentyű az aktuális panel valamely tételéhez, mint azonosító karakter van hozzárendelve, akkor az adott tétel kiválasztását jelenti.

### 2.5. A paneltétel azonosítása a tétel nevével

A panelrendszer egyes tételeit bizonyos esetekben egy univerzális névvel kell azonosítani.

Ez az azonosítás a következő esetekben szükséges:

- az egy tételre történő direkt ráugrás (F10, generálási és futtatási fázis),
- a helpfile tételeinek azonosításához,
- a DOS végpontok GOTO és PNLEXEC parancsainál,
- a futtatórendszer rutinjainak paramétereként.

Ha a panelrendszerben több azonos nevű tétel szerepel, akkor a

tétel nevét az előző szintű panelnévvel és egy "/" jellel kell kezdeni. Például, ha létezik a "Törlés" funkció a "Könyvtárak" és az "Adatállományok" nevű paneleken, akkor a két törlési funkció a "konyvtarak/torles" vagy az "adatallomanyok/torles" névvel azonosítható. Az egyes tételnevekből mindig elegendő annyit megadni, amennyi az egyértelmű megkülönböztetést lehetővé teszi. Az előbbi példában (az egyéb nevektől függően) elegendő lehet a "k/t" és az "a/t" nevekkal történő azonosítás.

A tétel keresésekor a rendszer a kis- és nagybetűk között nem tesz különbséget, és az ékezetes betűket a megfelelő ékezet nélküli betűként kezeli. (Az ékezetes nevek megadását l. a következő pontban).

Ha a panelrendszerben van duplikált alpanel, akkor annak tételeit az univerzális név sem azonosítja egyértelműen.

## 2.6. A paneltételek definiálása

Egy panel létrehozása az első tételének meghatározásával kezdődik. Egy tétel az alábbi három típus valamelyike lehet:

1. következő szintű alpanel további tételekkel,
2. C függvényt hívó tétel (végpont),
3. DOS parancssorozatot (Batch-et) végrehajtható tétel (végpont).

A CREAP által létrehozott panelrendszer első szintje kötelezően egy tétéleket tartalmazó panel (tehát nem lehet 2. vagy 3. típusú).

Egy tétel meghatározása a tétel nevének megadásával és típusának kiválasztásával történik.

A név megadása a képernyőn megjelenő ablakban lehetséges. Amennyiben a felhasználó úgy dönt, hogy az új tételre mégsincs szüksége, az ESC billentyű leütésével visszatérhet az előző panelszintre.

A név megadásakor ékezetes betűket is definiálhatunk úgy, hogy a betű előtt megadjuk a három lehetséges ékezet típus (' : ") közül a megfelelőt (pl.: a "körülmény" szó megadása a "k:or:ulm'eny" karaktersorozattal lehetséges). Mivel a teljes magyar betűkészlet a legtöbb számítógépen nincs meg, ezért a rövid és a hosszú magánhangzók egy része egyformán jelenik meg.

A tétel típusa a képernyő jobb felső sarkában megjelenő három típus (Panel, C function, DOS) valamelyikére való ráállással és az ENTER billentyű leütésével, vagy a P-C-D kezdőbetűk valamelyikének leütésével választható ki. A típus kiválasztásának visszautasítása (ESC) a tételdefiniálás visszavonását jelenti.

Ha egy új panel létrehozásakor a felhasználó visszautasítja az első tétel meghatározását, akkor a panel definiálatlan marad és a

rendszer visszatér az előző panelszintre.

## 2.7. Új paneltétel beszúrása

Egy panel tételeit két módon lehet bővíteni:

- új tétel beszúrásával az aktuális (kiemelt színnel jelölt) tétel elé (INS billentyű),
- új tétel hozzáadásával a tételekhez (Ctrl-A billentyű),

Az új tétel meghatározása az előző pontban leírt módon történik.

## 2.8. A paneltétel törlése

Az aktuális tétel törlése a DEL billentyűvel lehetséges. Egy tétel akkor törölhető, ha

- panel típusú, és nincs egy tétele sem,
- C függvényt hívó, és a hívott függvény nem definiált, (azaz az "undefined" függvényre van állítva)
- DOS típusú, és egyetlen DOS parancsa sincs definiálva.

Ha a tétel törölhető, akkor a rendszer biztonsági okból egy "Are You sure?" (Biztos ebben?) kérdéssel ellenőrzi a szándékosságot. Az erre adott 'Y' (igen) válasz hatására a kijelölt tétel törlődik.

Ha a törölni kívánt tétel nem "üres", akkor a törléshez a következőt kell elvégezni:

- C függvényt hívó tételnél a hívott függvényt az első, "undefined" függvényre kell állítani;
- DOS típusú tételnél a tétel összes DOS parancsát törölni kell;

- alpanel esetén az alpanel összes tételét törölni kell.

## 2.9. Alpanelek kimentése és beolvasása

Az alpanelek kimentése és beolvasása két különböző célt szolgálhat.

Egyrészt a monitorok fejlesztésekor az egyes monitorok gyakran tartalmaznak azonos alpaneleket (például segédprogramokat tartalmazó panel vagy konfigurációs panel), amelyek átmásolása az egyik panelrendszerből a másikba e funkciók segítségével lehetséges.

A másik felhasználási terület az lehet, amikor a tervezési fázisban az egyes alpanelek különböző változatai készülnek el, majd a végső változatba a legmegfelelőbb verzió épül be.

Az alpanel kimentése a következő módon történik:

- ráállás a kimentendő tételre (tehát a főpanel nem menthető ki alpanelként),
- az Alt-W (Write) billentyű leütésére a képernyő jobb felső sarkában megjelenik a file-kiválasztó menü, amely a létező panelfile-ok neveit és a "New file" tételeket tartalmazza,
- a "New file" választásakor a külön kérdésre meg kell adni az új file nevét, a kimentés erre automatikusan végrehajtódik,
- létező file kiválasztása esetén a rendszer az "Overwrite existing file?" (Felülírjuk a létező file-t?) kérdéssel ellenőrzi a szándékosságot. 'Y' vagy 'y' (igen) válasz esetén megtörténik a kimentés,
- a kimentés letiltható bármely kérdésre adott ESC leütéssel.

Az alpanel beolvasása a következő módon történik:

- ráállítás a kívánt tételre  
(Csak üres alpanelbe történhet a beolvasás, tehát ha egy már kitöltött alpanelt kívánunk felülírni, akkor annak tételeit a beolvasás előtt törölni kell!),
- az Alt-R (Read) billentyű leütésére a képernyő jobb felső sarkában megjelenik a létező panelfile-ok menüje,
- a kiválasztás után a tétel a kiválasztott file-ból feltöltődik,
- a beolvasás a kiválasztáskor leütött ESC billentyűvel lemondható.

## 2.10. Globális alpanelek

A panelrendszeren belül lehetőség van négy globális, azaz bárholonnan hívható paneltétel definiálására. Ezek mind generálási, mind pedig futtatási fázisban rendre az F5, F6, F7 és F8 billentyűkhöz vannak rendelve.

A globális paneltétel definiálása a megfelelő funkcióbillentyű leütésével kezdődik.

A funkcióbillentyű leütése után a következő esetek lehetségesek:

- ha a globális paneltétel még nincs definiálva:
  - a "Read subpanel from file?" (File-ből olvassuk be az alpanelt?) kérdés jelenik meg. Az erre adott 'Y' (igen) válasz esetén az Alt-R (l. az "Alpanelek kimentése és beolvasása" c. pontban) billentyű hatásával megegyező beolvasási művelet hajtódik végre.
  - No (nem) válasz esetén a panel definiálása az "A paneltételek definiálása" című pontban leírtak szerint történik.
- ha a globális paneltétel üres (törölhető), tehát nincs tétele

vagy nem definiált az általa hívott C függvény, vagy nincs definiált DOS parancsa, akkor megjelenik a "Do You want to delete this subpanel?" (Akarja törölni ezt az alpanelt?) kérdés. Az erre adott 'Y' igenlő válasz hatására az adott funkcióbillentyű globális paneltétele definiálatlanná válik. No (nem) válasz esetén a globális paneltétel a normál panelek mintájára módosítható.

- egyébként a globális paneltétel a normál panelek mintájára módosítható.

Mind generálási, mind pedig futtatási időben egyszerre csak egyetlen globális paneltétel lehet aktív, tehát ha valamelyik (panel típusú) globális paneltételt behívtuk a funkcióbillentyűje segítségével, akkor mindaddig, amíg e panelből ki nem léptünk, a globális paneltételt azonosító funkcióbillentyűk leütésére a rendszer figyelmeztető üzenetet ad, és a globális paneltétel behívását visszautasítja.

## 2.11. Aktív panelek definiálása

Gyakran van szükség egy panelág végrehajtása előtt és után valamilyen művelet (számítás, inicializálás...) elvégzésére.

Többek között erre ad lehetőséget az aktív panel definiálása. Egy panel a kezdő és befejező C függvények definiálásával tehető "aktív".

### 2.11.1. Kezdő C függvény meghatározása

Egy panel kezdő C függvénye a panel végrehajtása előtt minden esetben meghívódik (tehát az F10 billentyűs direkt ráúgrásnál is!) a következő két paraméterrel:

- a panel sorszáma az előző panelszinten (menün),
- a panelt leíró adatterület címe, a futtatórendszer rutinjainak átadható paraméterként.

A futtatási fázisban, ha e függvény visszatérési értéke 0, akkor a panel nem hajtódik végre. Ezzel a futtatási időben dinamikusan letiltható vagy engedélyezhető egy-egy aktív panel végrehajtása.

Az aktív panel kezdő C függvénye az Alt-S (Start C function) billentyű leütése után definiálható. A megfelelő C függvény meghatározása (illetve törlése) a "C függvényt hívó végpont" c. pontban leírtak szerint történhet.

Ha a főpanel kezdő C függvényét megadjuk, akkor ennek viselkedése a fentiekől eltérő lesz (l. az "Elszámoló rutinok beépítése" c. pontban).

### 2.11.2. Befejező C függvény meghatározása

Egy panel befejező C függvényére a panel végrehajtása végén kerül a vezérlés két paraméterrel:

- a panel sorszáma az előző panelszinten (menün),
- a panelt leíró adatterület címe, a futtatórendszer rutinjainak átadható paraméterként.

Az aktív panel befejező C függvénye az Alt-E (End C function) billentyű leütése után definiálható. A megfelelő C függvény meghatározása (illetve törlése) a "C függvényt hívó végpont" c.

pontban leírtak szerint történhet.

Ha a főpanel befejező C függvényét megadjuk, akkor ennek viselkedése a fentiektől eltérő lesz (l. az "Elszámoló rutinok beépítése" c. pontban).

## 2.12. Elszámoló rutinok beépítése

A főpanel kezdő és befejező C függvénye a fentiektől eltérően nem az aktív panel definiálásához, hanem elszámolási információk gyűjtéséhez alkalmas. Ezek a rutinok nem a panel végrehajtásának elején és végén hívódnak meg (hiszen a főpanel végrehajtása előtt és után a felhasználó bármit tehet), hanem a DOS parancssorozatok és a képernyőt nem védő C függvények végrehajtása előtt és után (a C függvények ilyen jellegű megkülönböztetését l. a "C függvényt hívó végpont" c. pontban). Így a két rutin alkalmas elszámolási információk gyűjtésére. A képernyőt védő C függvényeket úgy tekintjük mint elszámoláson kívüli funkciókat, így az ilyen végpontok esetén ez a két rutin nem hívódik meg. Ha ez mégis szükséges, akkor a kérdéses C függvénybe a programozó tetszés szerinti elszámolást építhet be (pl. explicit módon meghívhatja az elszámoló rutinokat).

### 2.13. Panelek átkötése, ismétlése

Bizonyos esetekben egyes menük, alpanelek a panelrendszeren belül azonos szerkezetűek. Ilyenkor nincs szükség az egyes ágakon a megegyező alpanelek fizikai megismétlésére, mivel a CREAP lehetőséget biztosít egy alpanel több ágon való elérésére.

Az átkötést (duplikálást) a következő példa teszi érthetővé:

Képzelt adatfeldolgozó rendszerünk több adatállományon értelmezi a

"Felvitel",  
"Módosítás",  
"Törlés" és  
"Listázás"

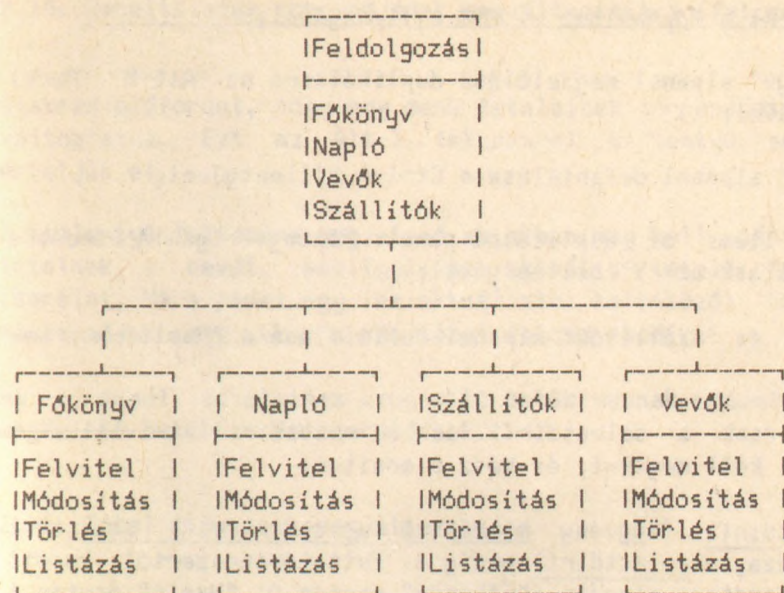
műveleteket.

A főpanel határozza meg, hogy mely adatállományon kell dolgozni, pl.:

Főkönyv  
Napló  
Vevők  
Szállítók

Az egyes műveleteket az adatállományokat tartalmazó (különböző) könyvtárakban lévő azonos nevű programok végzik. A megfelelő könyvtár kiválasztása tehát a főpanel egy tételének kiválasztásával történik.

Az elképzelt panelszerkezet a következő:



Ezek után a panelrendszer definiálása a következő lépésekben történik:

- A CREAP.EXE indítása;
- A "Főkönyv" tétel (alpanel) definiálása "Panel" típusal;
- Belépés a "Főkönyv" alpanelbe az ENTER billentyűvel;
- A "Felvitel" tétel definiálása DOS típusal; a DOS parancs: FELVITEL;
- A további három DOS végpont definiálása a megfelelő programnevekkel;
- A panel kezdő C függvényének meghatározása az Alt-S (Start C function) billentyűvel; a függvény: selectdir();
- A panel befejező C függvényének meghatározása az Alt-E (End C function) billentyűvel (a függvény: resetdir);

- Visszatérés a főpanelhez az ESC billentyűvel;
- A "Főkönyv" alpanel megjelölése duplikálásra az Alt-M (Mark) billentyűvel;
- A "Napló" alpanel definiálása a Ctrl-A billentyűvel;
- A "Use items of duplicated panel: Főkönyv? (y/n)" kérdésre igenlő válasz az 'Y' billentyűvel;
- A "Vevők" és "Szállítók" alpanel definiálása a "Napló"-hoz hasonlóan.

Ezután már csak a `selectdir()` és a `resetdir()` triviális C függvényeket kell megírni, és kész a monitor.

A `resetdir()` függvény mindössze a `chdir("..")` utasítást tartalmazza, a `selectdir()` pedig a futtatórendszerrel kapott első paramétere alapján ("Főkönyv" esetén 0, "Napló" esetén 1 stb., tehát akár a könyvtárnevek táblázatához is szolgálhat indexként) szintén a `chdir()` standard C függvényt hívja a megfelelő könyvtár nevével.

Ha korábban egy alpanel több panel típusú tételre rákötöttünk és az egyik tételnél ezt meg akarjuk szüntetni, akkor ezt a következő lépésekkel tehetjük meg:

- Ráállítás arra a tételre, amelyről a duplikált alpanelt le akarjuk kötni;
- A Panel típus megváltoztatása az Alt-T, majd a C vagy a D betű leütésével;
- ESC billentyűvel a végpont meghatározásának visszautasítása;
- A Panel típus visszaállítása az Alt-T és a P betű leütésével.

#### Figyelem!

A duplikált alpanel tételei nem érhetők el univerzális névvel.  
(l. "A paneltétel azonosítása a tétel nevével" c. pontban)

#### 2.14. Paneltételek sorrendjének megváltoztatása (felcserélés)

Gyakran előfordul, hogy egy menü tételeinek sorrendjét meg kell változtatni. Ezt az Alt-X (eXchange) billentyű segítségével érhetjük el.

A billentyű leütésére megjelenő ablakban meg kell adni annak a tételnek a nevét, amellyel az aktuális tételt fel akarjuk cserélni. Ha a panel egy (az aktuálistól különböző) tételének a nevét adjuk meg, akkor a két tétel felcserélődik.

Ha a panel tételeihez azonosító betűk vannak rendelve, akkor a megfelelő betűk is felcserélődnek.

#### 2.15. Tétel típusának megváltoztatása

A fejlesztés során egy tételről kiderülhet, hogy nem megfelelő típusú (például egy gyors C függvényrel érdemes megvalósítani az eredetileg DOS parancssorozatra hagyott funkciót).

Egy tétel típusának megváltoztatása az Alt-T (Type) billentyűvel lehetséges.

A típus megváltoztatása akkor lehetséges, ha a tétel

- panel típusú, és nincs egy tétele sem,
- C függvényt hívó, és a hívott függvény nem definiált, (azaz az "undefined" függvényre van állítva)
- DOS típusú, és egyetlen DOS parancsa sincs definiálva.

A tétel új típusának meghatározása az új tétel típusának meghatározásával megegyező módon történik. (l. "A paneltételek definiálása" c. pontban)

## 2.16. Végpontok meghatározása

A panelrendszer generálásának fontos része a feladatnak megfelelő végpontok, azaz a nem panel típusú tételek definiálása, hiszen a panelrendszer működése e végpontok meghatározásával nyer értelmet.

Egy végpont a következő két típus valamelyike lehet:

- C függvényt hívó végpont,
- DOS parancssorozatot végrehajtó végpont.

## 2.17. C függvényt hívó végpont

A panelrendszer által használt C függvények egy táblázatba kerülnek. (Az egyes végpontok által hívott C függvényeket a táblázatban elfoglalt pozíciójuk sorszáma azonosítja.) Egy C függvényt hívó végpont létrehozásakor vagy módosításakor a képernyőn megjelenik a panelrendszer által már definiált C függvények táblázata, amelynek első tétele mindig az "Undefined" (nem definiált) tétel.

Egy új (C függvényt hívó) végpont alapértelmezésben a nem definiált ("Undefined") függvényt "hívja". Ennek módosítása két módon lehetséges:

- egy már létező C függvény kiválasztása a kurzorvezérlő billentyűk használatával és a kiválasztást követő ENTER billentyű leütésével.
- új C függvény definiálásával és az ENTER billentyű leütésével.

Új C függvény definiálásának két módja van:

- a Ctrl-A billentyűkombinációval új függvény hozzáadása a tábl-

lázathoz,

- beszúrás a kiválasztott függvény elé az INS billentyű leütésével.

Egy függvény neve meg kell, hogy feleljen a C nyelv konvencióinak és legfeljebb 15 karakter hosszú lehet.

Egy C függvény nevét megváltoztatható a kurzorbillentyűkkel való ráállítás, majd az F2 billentyű leütése után.

Figyelem! Futtatási időben a függvény nevének első karaktere határozza meg a függvény hívási módját:

ha az első karakter a '\_' (aláhúzás), akkor panelrendszer futtatásakor (a függvény hívásakor) a képernyőn lévő információk mentéséről a futtatórendszer nem gondoskodik. Ez akkor célszerű ha

a C függvény a képernyő tartalmára nincs hatással,  
vagy ha

a C függvény a képernyő tartalmát felhasználja, de változatlan formában visszaállítja.

ha az első karakter nem a '\_', akkor a futtatórendszer a képernyőn lévő paneleket (ablakokat) lezárja, majd a függvény végrehajtása után visszaállítja az eredeti állapotot. Ilyenkor a futtatórendszer a C függvényt, mint a képernyőt védő függvényt kezeli.

Ha egy végpontot egy másik C függvényre állítunk és az általa eddig hívott függvényre máshonnan nem hivatkozunk, akkor a végpontdefiniálásból való kiszálláskor a rendszer automatikusan felkínálja a C függvény törlését az őld C function "X" not used. Delete from table? (y/n)" (A régi X függvény nem használatos. Töröljük a táblázatból?) kérdéssel. Igenlő ('Y') válasz esetén a rendszer a függvényt törli a panelrendszerből.

Ha egy már definiált C függvény szükségtelennek bizonyul, akkor minden rá vonatkozó hivatkozást törölni kell (az ezen függvényt

hívó végpontokat az "Undefined"-re kell állítani), majd a függvény nevére való ráállítás után a DEL billentyű leütésével törölni kell.

A meghívott C függvény paraméterei futtatáskor:

1. a tétel panelen belüli sorszáma (int),
2. a hívó panelt azonosító terület címe (char \*). E paraméter használata opcionális, a futtatórendszer szolgáltatásait végző rutinok paramétereként adható tovább (l. "A futtatórendszer panelkezelő függvényei" c. pontban).

A C függvénnyel megvalósított végpontot jól szemléltetik a könyv végén található "Példa a CREAP futtatórendszer használatára" c. fejezet "Utilities" alpaneljéhez tartozó végpontok.

## 2.18. DOS parancssorozatot hívó végpont

A végpontok másik típusába a DOS parancssorozatot indító végpontok tartoznak. A DOS parancssorozat tetszőleges számú parancsból állhat. Egy parancs tartalmazhat

- egy DOS-ban érvényes tetszőleges parancsot vagy
- egy, a CREAP futtatórendszere által kezelt 'kvázi' Batch parancsot.

Az egyes parancsok ezenkívül tartalmazhatnak szimbolikus paramétereket is.

A parancssorozatok megadása az ilyen típusú végpontra való ráállással kezdődik.

Ekkor a képernyőn megjelennek az eddig definiált sorok. A parancssorok kezelése a következő:

- kiválasztás (szokásos módon): kurzorvezérlő billentyűkkel,

- új sor beszúrása az aktuális (kiemelt) sor elé: INS billentyű,
- új sor hozzáadása a létező sorokhoz: Ctrl-A billentyű,
- létező sor törlése: ráállás, majd a DEL billentyű leütése,
- létező sor módosítása: ráállás, majd az ENTER billentyű leütése,
- a CREAP Batch nyelvi bővítéseinek leírása: F1 billentyű,
- a már definiált szimbolikus paraméterek rendezett listája: F2 billentyű.

A panelrendszer futtatásakor a DOS végpont végrehajtása előtt a futtatórendszer a képernyőn lévő paneleket (ablakokat) lezárja, majd a végponthoz tartozó parancsok végrehajtása után visszaállítja az eredeti állapotot.

Futtatási fázisban az egyes parancssorok végrehajtás előtt megjelennek a képernyőn, kivéve, ha a parancssor első karaktere a '@' karakter.

A DOS parancssorozattal megvalósított végpontokat jól szemlélteti a könyv végén található "Példa a CREAP futtatórendszer használatára" c. fejezet "compile & link" funkciója.

### 2.18.1. A szimbolikus paraméterek

A CREAP futtatórendszer értelmezi a parancsokban a szimbolikus paramétereket és végrehajtáskor ezek helyére azok aktuális értékét helyettesíti.

A szimbolikus paraméter formája:

\$(név)

ahol a név egy legfeljebb 10 karakterből álló alfanumerikus azonosító.

A generálási fázisban a már definiált szimbolikus paraméterek rendezett listája a végpont módosításakor az F2 billentyűvel kérhető le.

Futtatási fázisban a szimbolikus paramétereknek értéke

- módosítható (F2),
- beolvasható file-ból (F3),
- kimenthető file-ba (F4).

Ezáltal egy monitor különböző paraméterfile-ok segítségével (több paraméterérték halmazzal) több környezetre is alkalmassá tehető. (Például szimbolikus paraméterekkel jelöljük a konfigurációt, adat- és programkönyvtárakat stb.)

Megjegyzés: a szimbolikus paramétereket a program C függvényei is elérhetik a futtatórendszer PNL\_getarg() és PNL\_setarg() rutinjaival.

## 2.18.2. A kiterjesztett DOS parancsnyelv

A CREAP futtatórendszere értelmezi a DOS parancsait, és ezen kívül értelmezi a következőkben leírt nyelvi elemeket is.

### 2.18.2.1. ASK

Formája:

ASK <szöveg> <parancs>

ahol <szöveg> egy idézőjelek közé zárt karaktersor, vagy egy olyan karaktersor, amelyben nincs szóköz,

<parancs> egy tetszőleges parancs (amely lehet egy GOTO is).

Funkciója:

- hatására egy ablakban megjelenik a <szöveg>, és a rendszer egy igen/nem válaszra vár,
- a válaszadás után visszaáll az eredeti képernyő,
- igen ('Y' vagy 'y' - magyar változatnál 'I', 'i') válasz esetén a <parancs> végrehajtódik,
- nem ('N', 'n' vagy ESC) válasz esetén a <parancs> nem hajtódik végre.

### 2.18.2.2. BEEP

Formája:

BEEP <kezdő> <vég> <hossz>

ahol <kezdő> a hangjelzés kezdőfrekvenciája Hz-ben,

<vég> a hangjelzés végfrekvenciája Hz-ben,

<hossz> a hangjelzés hossza 1/100 s-ban.

Funkciója:

- a megadott <kezdő>-hangtól a <vég>-hangig hangjelzést ad <hossz> ideig.

### 2.18.2.3. EXIT

Formája:

EXIT

Funkciója:

- befejezi (megszakítja) a parancssorozat végrehajtását (a DOS parancssorozat további parancsai nem hajtódnak végre).

### 2.18.2.4. GOTO

Formája:

GOTO <cél>

ahol <cél> egy parancssor sorszám vagy egy paneltétel neve.

Funkciója:

- az utasítás hatására a vezérlés a  
a megadott sorszámú parancssorra adódik,  
vagy  
a megadott paneltételre áll rá, de annak végrehajtását nem  
kezdi el.

(A tétel nevének megadását l. még "A paneltétel azonosítása a  
tétel nevével" c. pontban).

#### 2.18.2.5. IF

Formája:

IF <szint> <parancs>

ahol <szint> egy hibaszint,

<parancs> pedig egy Batch parancs.

Funkciója:

- ha az előzőleg végrehajtott parancs hibaszintje a megadott  
<szint>-tel egyenlő vagy annál nagyobb, akkor a <parancs>  
végrehajtódik.

Ha az IF utasítás után nem számjegy áll, akkor a DOS operációs  
rendszer IF parancsa hajtódik végre.

#### 2.18.2.6. IFN

Formája:

IFN <szint> <parancs>

ahol <szint> egy hibaszint,

<parancs> pedig egy Batch parancs.

Funkciója:

- ha az előzőleg végrehajtott parancs hibaszintje kisebb a megadott <szint>-nél, akkor a <parancs> végrehajtódik.

#### 2.18.2.7. PAUSE

Formája:

PAUSE <szöveg>

ahol <szöveg> egy idézőjelek közé zárt karaktersor, vagy egy olyan karaktersor, amelyben nincs szóköz.

Funkciója:

- kiírja a <szöveget> egy ablakba, és egy ESC leütéséig felfüggeszti a parancsok végrehajtását. Az ESC leütése után visszaáll az eredeti képernyő.

#### 2.18.2.8. PNLEXEC

Formája:

PNLEXEC <panel>

ahol <panel> egy paneltétel neve.

Funkciója:

- az aktuális paneltétel végrehajtását abbahagyja, majd a megadott tétellel folytatja (a tétel nevének megadását l. "A paneltétel azonosítása a tétel nevével" c. pontban).

## 2.19. Az ablakok paramétereinek meghatározása

### 2.19.1. Az ablakok helyének meghatározása

Egy panel ablaka a létrehozásakor a képernyő közepére kerül.

Ennek megváltoztatására a Ctrl-lal kombinált kurzorvezérlő (numerikus billentyűzet) billentyűkkel van lehetőség:

az ablak mozgatása balra:

Ctrl-<balra nyíl> (Ctrl-4)

az ablak mozgatása jobbra:

Ctrl-<jobbra nyíl> (Ctrl-6)

az ablak mozgatása fölfelé:

Ctrl-<fölfelé nyíl> (Ctrl-8) vagy Ctrl-Home vagy Ctrl-PgUp

az ablak mozgatása lefelé:

Ctrl-<lefelé nyíl> (Ctrl-2) vagy Ctrl-End vagy Ctrl-PgDn

A Pull-down menük esetén (mivel ezek mindig a teljes képernyőszélességet elfoglalják) csak a függőleges mozgatásra van lehetőség.

Az ablakok mozgatása mind generálási, mind pedig futtatási fázisban lehetséges, de futtatási fázisban a mozgatás csak egy futási menetre érvényes, míg generálási fázisban a megváltoztatott pozíció megőrződik.

## 2.19.2. Az ablakok méretének módosítása

Egy ablak szélességét annak létrehozáskor a panel címének hossza határozza meg, magassága alapértelmezésben 7 sor. Ezek megváltoztatása a következő billentyűkkel lehetséges:

az ablak magasságának növelése:

Shift-<lefelé nyíl> (Shift-2)

az ablak magasságának csökkentése:

Shift-<felfelé nyíl> (Shift-8)

az ablak szélességének növelése:

Shift-<jobbra nyíl> (Shift-6)

az ablak szélességének csökkentése:

Shift-<balra nyíl> (Shift-4)

Az ablakméret csak generálási fázisban változtatható.

Ha egy panel tételei nem férnek el az ablakban, akkor az ablak megjelenítése a következő formára módosul:

- a tételnevek előtt (egy függőleges vonallal elválasztva) egy üres oszlop képződik,
- ha az ablakban a legfelső tétel nem a panel első tétele, akkor ezt az üres oszlop felső pozíciójában egy felfelé nyíl jelzi,
- ha az ablakban a legalsó tétel nem a panel utolsó tétele, akkor ezt az üres oszlop alsó pozíciójában egy lefelé nyíl jelzi.

### 2.19.3. Az ablakok színeinek meghatározása

A panelrendszerben 5 szín használatos, amelyből három panel-specifikus, kettő pedig globális.

Az aktuális panelre jellemző színek:

- Alapszín (base color): a panel alapszíne a keret és a tételnevek kiírásához.
- Kiemelt (highlighted) szín: az aktuális tétel kiemeléséhez.
- Szerkesztési (edit) szín: futtatási fázisban a védett (letiltott) tételek színe (generálási fázisban a tételnevek megadásakor használt szín).

Globális színek:

- Üzenetek színe (message): hibajelzésekhez, üzenetekhez.
- Helpkiírási szín: a segítő információk kiírásához.

A színek beállítása az Alt-C (Colors) billentyű leütése után történhet. Ekkor a képernyőre kerül egy ablak, amelyben (magyarázaton kívül) az öt szín jelenik meg, mindegyiknél az előtér és háttér színek megjelölésével (foreground/background). A módosítandó mezőre a kurzorvezérlő billentyűkkel lehet ráállni. Magát a színt a + és - (plusz/minusz) billentyűkkel lehet a kívánt változatra beállítani. A változtatásokat az ENTER billentyűvel lehet lezárni, vagy az ESC billentyűvel eltörölni.

#### 2.19.4. A keret típusának állítása

A CREAP panelrendszerében a következő kerettípusok ismertek:

- dupla vonal,
- szimpla vonal,
- vízszintesen dupla, függőlegesen szimpla vonal,
- vízszintesen szimpla, függőlegesen dupla vonal,
- szóközők (üres keret).

A megfelelő kerettípus kiválasztása az Alt-F (Frame) billentyűvel hívható. Erre a képernyőn megjelenik a keretmintákat tartalmazó ablak, ahol a Left (balra nyíl) és Right (jobbra nyíl) billentyűk segítségével kiválasztható a kívánt kerettípus. Az ENTER billentyű hatására a változtatás rögzíthető, ESC hatására az eredeti keret marad.

#### 2.20. A tételek nevének módosítása

Egy tétel neve az Alt-A (Actual) billentyű leütése után módosítható.

A legfelső panel (főmenü) neve az Alt-N (Name) billentyű leütése után módosítható.

## 2.21. Pull-down menük

A CREAP az ablakos menük mellett lehetőséget ad Pull-down menük használatára is.

Egy ablakos menü az Alt-P (Pull down) billentyű segítségével állítható át Pull-down menüvé (és vissza). Ez akkor lehetséges, ha a panel tételeinek nevei elférnek egy sorban (a megfelelő keret és elválasztó szóközzel együtt).

Az ablakos panelekre vonatkozó műveletek a generálási fázisban a Pull-down menükre általában nem érvényesek; a művelet előtt a Pull-down menüt ablakos panellá kell átalakítani, szintén az Alt-P billentyű leütésével.

Futtatási fázisban a Pull-down menük kezelése értelemszerű.

## 2.22. A paneltételeket azonosító karakterek meghatározása

Az egyes panelek tételeinek kiválasztása a kurzorvezérlő billentyűk és az ENTER használatán kívül a tételazonosító karakterek segítségével is lehetséges. Ezek definiálása, illetve módosítása az Alt-L (Letter) billentyű leütésével valósítható meg.

Az Alt-L billentyű hatására a képernyőn egy ablak jelenik meg a "Sequence of the select letters" (A kiválasztó betűk sorozata) felirattal. Ez után lehet beütni a tételek sorrendjének megfelelően az egyes tételeket azonosító karaktereket. Ennek lezárása (ENTER) után a "Do You want to capitalize the panel items?" (Nagybetűssé akarja tenni a panel tételeit?) kérdés jelenik meg. Ha erre 'Y' vagy 'y' (igen) a válasz, akkor a tételnevek kisbetűsekké válnak, kivéve az azonosító karaktert (természetesen csak akkor, ha ez betű).

Ha egy panel tételeihez azonosító karakterek vannak rendelve, akkor egy azonosító karakter leütése egyenértékű (csak sokkal gyorsabb) a kurzorvezérlőkkel való ráállás és az ENTER billentyű leütésével történő végrehajtással.

## 2.23. A helpfile illesztése a panelrendszerhez

A CREAP segítségével készült rendszerekhez a segítő információk illesztése két lépésben történik:

- a helpfile létrehozása egy tetszőleges szövegszerkesztővel,
- a CREAP Alt-H (Help indexing) billentyűjével a helpfile indexelése és ennek beépítése a panelrendszerbe.

A helpfile felépítése:

A helpfile-ban az egyes tételeket a Form-feed (lapemelés, 12 ASCII, Ctrl-L) karakter és a tétel neve azonosítja (l. még "A paneltétel azonosítása a tétel nevével" c. pontban).

A segítő információ sorainak hossza legfeljebb 78 karakter lehet.  
Példa:

```
-----  
|   Menü   |  
-----  
| Kódolás |  
| Fordítás |  
| Szerkesztés |  
| Futtatás |  
-----
```

A fenti panelhez tartozó helpfile a vonalak közötti rész:

^L

Kódolás

A World Star szövegszerkesztőt  
hívja az ACTFILE aktuális  
értékevel, .C típussal

^L

Fordítás

A Microsoft C V4.00 fordítót  
hívja az ACTFILE aktuális  
értékevel, debug opcióval

^L

Szerkesztés

A Microsoft C V4.00 szerkesztőt  
hívja az ACTFILE aktuális  
értékevel

^L

Futtatás

Futtatja az aktuális ACTFILE-t

A generálási fázisban a help illesztésekor kell megadni a helpfile nevét, a korábban már leírt file-név kiválasztási módszerrel. Amennyiben az Alt-H leütése után a helpfile kiválasztását az ESC billentyűvel visszautasítjuk, akkor a panelrendszer helpfunkciói megszűnnek.

Fontos figyelmeztetés! Ha a helpfile akár egy karakterrel is módosul, akkor a panelrendszert aktualizálni kell az Alt-H funkcióval!

#### 2.24. Kezelési segédlet megjelenítése

Az Alt-F1 vagy a "?" billentyű hatására a képernyőn megjelennek:

- a kezelési információk, azaz a különböző vezérlőbillentyűk funkciói,
- az átkötésre (ismétlésre) megjelölt panel neve,
- az F5-F8 billentyűkhöz rendelt globális paneltételek nevei.

Futtatási fázisban a kezelési segédlet annyiban különbözik, hogy csak az adott pillanatban érvényes billentyűk magyarázata jelenik meg a képernyőn.

#### 2.25. A panelszerkezet megjelenítése és dokumentálása

A panelszerkezet áttekintésére az Alt-D (Display) billentyű szolgál. Hatására a képernyőn megjelennek:

- a tételnevek a beágyazási szintnek megfelelően paragrafusozva,
- a tétel típusa: P (panel), C (C függvény) vagy D (DOS),
- van-e help a tételhez (+/-),
- panel típus esetén a kezdő és befejező C függvény neve,
- C függvény típus esetén a hívott C függvény neve,
- DOS típus esetén az első Batch sorból a képernyőn kiferő rész.

A kiírás az ESC billentyű leütésével megszakítható.

A képernyőre való kiírás után megjelenik a "Do You want printable documentation? (y/n)" (Akar nyomtatható dokumentációt?) kérdés.

'Y' (igen) válasz esetén egy ablakban az "Enter the documentation file name:" (Adja meg a dokumentáció file nevét) kérdésre meg kell adni a kívánt file nevét.

A dokumentációs file-ban a képernyőn megjelenő (fent említett) információkon kívül minden panel típusú tételről rajz készül, és kiíródik a tételekhez tartozó help szövege.

A dokumentációs file fejlécezett, számozott lapokra osztott.

A nyomtatható dokumentációt a "Példa a CREAP futtatórendszer használatára" c. pontban szemléltetjük.

## 2.26. Kilépés a DOS-ba

A felhasználó az F9 billentyű hatására mind generálási, mind pedig futtatási fázisban kiléphet a DOS operációs rendszerbe. Az operációs rendszerből az "exit" paranccsal lehet a CREAP-ba vagy a futtatott panelrendszerbe visszatérni. Mivel a DOS-ba való kilépés egy DOS másolat végrehajtása, ezért nagy programok esetében előfordulhat, hogy a meghívott DOS-ban bizonyos nagyobb memóriaigényű programok nem futnak le a memóriahiány miatt.

## 2.27. A CREAP verzió kiírása

Az Alt-V (Version) billentyű hatására a képernyő közepén kiírásra kerül az aktuális CREAP változat azonosító szövege.

### 3. A futtatórendszer használata

A további fejezetekben a CREAP környezet futtatórendszerével foglalkozunk. A futtatórendszer használata két módon lehetséges, mint ahogy azt a "Bevezetőben" már leírtuk.

Ha a panelrendszer nem tartalmaz C függvényekre való hivatkozásokat, azaz csak DOS parancssorozatokkal megvalósított funkciókat tartalmaz, akkor a futtatórendszer használata csak a PANEL.EXE vagy a PANELM.EXE programok használatát jelenti. A futtatórendszer (ha a +p opció nem szerepel) a panelfile nevét az elindított program nevéből képi. Így ha a csak DOS parancssorozatokot használó panelrendszer a LETAR.PNL file-ban van, akkor a futtatás a következő módon könnyíthető meg:

Készítsünk a PANEL.EXE (vagy a PANELM.EXE) programról egy másolatot a LETAR.PNL-t tartalmazó könyvtárba, LETAR.EXE néven.

Ezután a program egyszerűen a >LETAR paranccsal indítható.

Abban az esetben ha a szimbolikus paraméterek file-ja a LETAR.ARG, akkor az argumentum file automatikusan betöltődik.

(l. még "A futtatórendszer operátori kezelése" c. pontban.)

Amennyiben a létrehozandó szoftver funkciói nem valósíthatók meg kizárólag DOS parancssorozatokkal, akkor ezeket a funkciókat C nyelven megírt függvényekkel kell lefedni. Ezek lehetnek:

- a panelrendszer végpontjai,
- aktív panelek kezdő és befejező funkciói,
- elszámolást végző rutinok.

A futtatórendszer a PANEL.LIB (vagy magyar nyelvű változata: a PANELM.LIB) tárgykódkönyvtárban helyezkedik el. Beépítése a felhasználó számára az összeszerkesztésen kívül semmilyen tevékenységet nem igényel.

A felhasználói program a következő módon épül fel:

```
main(argc,argv)
  int argc;
  char *argv[];
  {
    inicfuncs();
    argc = inipanel(argc,argv);
    .
    .
    .
    dopanel();
    exit(0);
  }

userfunc1(..)
  {
  }

userfunc2(..)
  {
  }
.
.
```

Az inicfuncs() függvényt a CREAP által generált .CP kiterjesztésű forrásfile tartalmazza. A forrásmodult vagy le kell fordítani és hozzászerkeszteni a főprogram tárgymoduljához, vagy egy #include utasítással be kell illeszteni a főprogram forrásprogramjába. Ez a függvény teremti meg a CREAP programmal definiált C függvénytáblázat és a futtatórendszer közötti kapcsolatot. Az inicfuncs() hívása meg kell, hogy előzze az inipanel() függvény hívását. Az inicfuncs() függvény csak egyszer hívható.

Az inipanel() függvény feldolgozza, majd törli a panelrendszerre vonatkozó programparamétereket, inicializálja a panelrendszert és visszaadja a megtisztított ARGV-re vonatkozó ARGC értéket. Az ARGC értékadásának természetesen csak akkor van értelme, ha a felhasználó saját paraméterkezelést végez, például a dopanel() hívása előtt a rendszer rááll a programparaméterként megadott könyvtárra. Az inipanel() függvény csak egyszer hívható.

A dopanel() végzi a panelrendszer "végrehajtását". A függvény visszatérése a főmenüből való kiszállást jelenti. A dopanel() függvény többször is hívható.

Megjegyzés:

A PANEL.EXE program (a fentiek alapján) a következő utasításokból áll (nincs szükség az inicfunc() függvényre):

```
main(argc,argv)
  int argc;
  char *argv[];
  {
    inipanel(argc,argv);
    dopanel();
  }
```

A PANEL.EXE és a PANELM.EXE programok a fenti forrást tartalmazó PANEL.C forrásprogramból a következő, Microsoft C környezetet feltételező utasításokkal reprodukálhatók:

```
>cl PANEL.C -link PANEL
>cl PANEL.C -FePANELM.EXE -link PANELM
```

Jól látható, hogy a két program létrehozása csak a felhasznált könyvtárban (és természetesen az .EXE file nevében) különbözik.

#### 4. A felhasználói C függvények

A következő függvénytípusok a felhasználó által megírandó és a futtatórendszer által meghívott függvények. Az egyes nevek természetesen nem kötelezőek, csak a funkcióra utalnak.

##### 4.1. A végpontot lefedő C függvény

###### Összefoglalás

userfunc(sernum,ptr)	
int sernum;	a tétel sorszáma a panelen
char *ptr;	a hívó panel pointere

###### Leírás

A függvény a C-ben írt végpont funkcióját valósítja meg. A paraméterek a futtatórendszer rutinjainak továbbadandó argumentumok, az első paraméter ezenkívül alkalmas arra is, hogy a hasonló funkciók egy függvénnyel lefedhetőek legyenek és a függvény egy táblázat vagy switch alapján végezze feladatát.

###### Visszatérési érték

Nincs.

###### Példa

Az alábbi példa egy hónap kiválasztására alkalmas. Az aktuális menü (ptr paraméter) 12 eleme rendre a hónapok nevei.

```
int month;
char *monthname;

select_month(sernum,ptr)
    int sernum;
    char *ptr;
    {
        char *PNL_name();      /* 5. fejezet */

        month = sernum;
        monthname = PNL_name(sernum,ptr);
        .
        .
        .
    }
}
```

#### 4.2. Az aktív panel kezdőfüggvénye

##### Összefoglalás

```
int startfunc(sernum,ptr)
    int sernum;          a tétel sorszama a panelen
    char *ptr;          a hívó panel pointer
```

##### Leírás

Az aktív panel kezdőfunkciója a menü megnyitása előtt meghívódik. Ennek alapján az adott panelág inicializálási, tételletiltási stb. funkciói elvégezhetők, avagy bizonyos esetekben a panelág végrehajtása visszautasítható.

A paraméterek a futtatórendszer rutinjainak továbbadandó argumentumok, az első paraméter ezenkívül alkalmas arra, hogy a hasonló funkciók egy függvénnyel lefedhetőek legyenek és a függvény egy táblázat vagy switch alapján végezze feladatát.

##### Visszatérési érték

- 1, ha a panelág végrehajtható,
- 0, ha a panelág végrehajtása visszautasítjuk.

### Példa

A példában az adott panelág csak egy jelszó megadása esetén hajtható végre (például egy könyvelési rendszerben a sztornó menüje). Feltételezzük, hogy a `check_passw()` függvény a menüben elfoglalt sorszám alapján eldönti, hogy a megadott jelszó az adott funkció végrehajtására jogosult-e. Az `init_branch()` függvény egy switch vagy táblázat alapján döntheti el, hogy például mely file-okat kell megnyitni.

```
start_and_check(sernum, ptr)
{
    int sernum;
    char *ptr;
    {
        if (check_passw(sernum) == 0)
            return(0);
        init_branch(sernum);
        return(1);
    }
}
```

### 4.3. Az aktív panel befejezőfüggvénye

#### Összefoglalás

```
int endfunc(sernum, ptr)
    int sernum;           a tétel sorszáma a panelen
    char *ptr;           a hívó panel pointera
```

#### Leírás

Az aktív panel befejezőfüggvénye a menüből való kiszálláskor hívódik meg. Ennek alapján az adott panelág lezárási, törlési stb. funkciói végezhetők el.

A paraméterek a futtatórendszer rutinjainak továbbadandó argumentumok, az első paraméter ezenkívül alkalmas arra, hogy a hasonló funkciók egy függvénnyel lefedhetőek legyenek és a függvény egy táblázat vagy switch alapján végezze feladatát.

### Visszatérési érték

Nincs.

### Példa

A példánk az előző pontban leírt példa párja. A `close_branch()` függvény egy switch vagy táblázat alapján döntheti el, hogy például mely file-okat kell lezárni.

```
endfunc(sernum,ptr)
  int  sernum;
  char *ptr;
  {
    close_branch(sernum);
  }
```

## 4.4. Az elszámolást végző rutinok

### Összefoglalás

<code>int start_acnt(sernum,ptr)</code>	
<code>int  sernum;</code>	a tétel sorszáma a panelen
<code>char *ptr;</code>	a hívó panel pointere
<code>int end_acnt(sernum,ptr)</code>	
<code>int  sernum;</code>	a tétel sorszáma a panelen
<code>char *ptr;</code>	a hívó panel pointere

### Leírás

A főpanelhez rendelt kezdő és befejező függvények a DOS és a képernyőt nem védő C végpontok végrehajtása előtt és után hívódnak meg, így alkalmasak elszámolási információk regisztrálására.

A paraméterek a futtatórendszer rutinjainak továbbadandó argumentumok.

Visszatérési érték

Nincs.

Példa

Az alábbi rutinpár a SYSLOG.DAT file-ba kiírja az aktuális operátor kódját, a funkció nevét és a végrehajtásának kezdő, illetve befejező idejét. A keletkezett file egy alkalmas segédprogrammal elszámolási táblázatot nyomtathat az egyes végpontok használatáról. A példában feltételezzük, hogy az operátort a főprogram (a dopanel() hívása előtt) azonosította és az operátor neve az actuser karaktertömbben van. A SYSLOG.DAT file-t szintén a főprogram nyitja meg, majd a dopanel() visszatérése után lezárja.

```
extern FILE *acntfile;
extern char actuser[];

acnt_on(sernum,ptr)
    int sernum;
    char *ptr;
    {
        long time();
        char *ctime();
        char *PNL_name();      /* 5. fejezet */
        char *actfunc;
        char *acttime;
        long tloc;

        actfunc = PNL_name(sernum,ptr);
        time(&tloc);
        acttime = ctime(&tloc);
        fprintf(acntfile,"START: %-20.20s %-20.20s %s\n",
                actuser,actfunc,acttime);
    }
}
```

```
acct_off(sernum,ptr)
int sernum;
char *ptr;
{
    long time();
    char *ctime();
    char *PNL_name();      /* 5. fejezet */
    char *actfunc;
    char *acttime;
    long tloc;

    actfunc = PNL_name(sernum,ptr);
    time(&tloc);
    acttime = ctime(&tloc);
    fprintf(acntfile,"END: %-20.20s %-20.20s %s\n",
            actuser,actfunc,acttime);
}
```

## 5. A felhasználó által használható függvények és változók

### 5.1. A futtatórendszer panelkezelő függvényei

Az alábbi pontokban a futtatórendszer azon függvényeit írjuk le, amelyek a panelrendszer használatát, vezérlését teszik programozhatóvá.

A futtatórendszer függvényeinek átadandó paraméterek közül magyarázatra szorulnak a "ptr", a "sernum" és az "str" névvel jelölt paraméterek.

A "ptr" paraméter mindig a felhasználói rutinnak a futtatórendszerrel kapott második paraméterét jelenti. Ez azonosítja az aktuális panelt a futtatórendszer részére. A programozó számára ez a paraméter nem értelmezhető, egyszerűen tovább kell adni a hívott futtatórendszerbeli rutinnak, ha az igényli.

A "sernum" paraméter a "ptr" paraméterrel azonosított panelen belüli indexet jelenti, 0-tól számozva.

Ha tehát az aktuális panel a következő:

Könyvelés
Naplózás
<u>Főkönyv</u>
Vevők
Szállítók
Mentés

és az aktuális tétel a "Főkönyv", amely egy C függvényt hív, akkor az előző pontban leírt "userfunc(sernum,ptr)" rutin kapott paramétereinek jelentése és értéke:

sernum: a tétel indexe a panelen, azaz 1,  
ptr : a "Könyvelés" panel azonosítója a futtatórendszer számára.

Az "str" paraméter egy tételnek a panelrendszeren belüli univerzális azonosítására szolgál (l. még "A paneltétel azonosítása a tétel nevével" c. pontban).

Itt hívjuk fel a figyelmet arra, hogy a futtatórendszer dinamikus lehetőséget ad a menük egyes tételeinek letiltására. Ezt más néven a védettség állíthatóságának is értelmezhetjük. A védettség ki, illetve bekapcsolását és a védettség lekérdezését 4 rutin teszi lehetővé:

PNL\_protect(), PNL\_isprotect, PNL\_glprotect() és  
PNL\_glisprotect().

#### 5.1.1. inipanel()

##### Összefoglalás

inipanel(argc,argv)	
int argc;	a programparaméterek száma
char *argv[];	a programparaméterek

##### Leírás

Inicializálja a panelrendszert. A programban csak egyszer hívható. A main program paramétereit kötelezően át kell adni.

##### Visszatérési érték

A futtatórendszernek szóló opcióktól megtisztított programparaméterek száma.

### 5.1.2. dopanel()

#### Összefoglalás

dopanel()

#### Leírás

Végrehajtja a panelrendszert. A függvény visszatérése a főmenü elhagyását jelenti. A rutin a programban többször is meghívható.

#### Visszatérési érték

Nincs.

### 5.1.3. PNL\_protect()

#### Összefoglalás

PNL\_protect(sernum, ptr, onoff)

int sernum;	az állítandó tétel indexe
char *ptr;	a kapott panelpointer
int onoff;	be- vagy kikapcsolás

#### Leírás

A függvény a paraméterekkel megadott menütétel letiltását kapcsolja be, illetve ki, a harmadik paraméter értékétől függően. Ha ez az érték 0, akkor a tétel nem lesz letiltva, egyéb érték esetén a tétel nem végrehajtható.

#### Visszatérési érték

Nincs.

#### 5.1.4. PNL\_isprotect()

##### Összefoglalás

```
PNL_isprotect(sernum,ptr)
    int sernum;          az állítandó tétel indexe
    char *ptr;          a kapott panelpointer
```

##### Leírás

Visszaadja, hogy a paraméterek által meghatározott tétel letiltott-e.

##### Visszatérési érték

0, ha a tétel nem letiltott,  
1 egyébként.

### 5.1.5. PNL\_glprotect()

#### Összefoglalás

PNL_glprotect(str,onoff)	
char *str;	az állítandó tétel neve
int onoff;	be- vagy kikapcsolás

#### Leírás

Az str stringgel meghatározott tétel letiltását kapcsolja be vagy ki a második paraméter értékétől függően. Ha ez 0, akkor a tétel nem lesz letiltott, egybként a tétel nem lesz elérhető (a tétel nevének megadását l. még "A paneltétel azonosítása a tétel nevével" c. pontban).

#### Visszatérési érték

-1, ha a nincs ilyen nevű tétel a panelrendszerben,  
0, ha a tétel ettől kezdve nincs letiltva,  
1, ha a tétel ettől kezdve letiltott.

### 5.1.6. PNL\_glisprotect()

#### Összefoglalás

PNL\_glisprotect(str)  
char \*str;                      a tétel neve

#### Leírás

Visszaadja, hogy az str névvel meghatározott paneltétel védett-e vagy sem (a tétel nevének megadását l. még "A panel-tétel azonosítása a tétel nevével" c. pontban).

#### Visszatérési érték

- 1, ha a nincs ilyen nevű tétel a panelrendszerben,
- 0, ha a tétel nincs letiltva,
- 1, ha a tétel letiltott.

### 5.1.7. PNL\_getarg()

#### Összefoglalás

char \*PNL\_getarg(aname)  
char \*aname;                      a szimbolikus paraméter neve

#### Leírás

Visszaadja a megadott nevű szimbolikus paraméter értékét.

#### Visszatérési érték

- 0, ha a szimbolikus paraméter értéke definiálatlan,  
a szimbolikus paraméter aktuális értéke egyébként.

### 5.1.8. PNL\_setarg()

#### Összefoglalás

PNL_setarg(aname,newval)	
char *aname;	a szimbolikus paraméter neve
char *newval;	a beállítandó érték

#### Leírás

A megadott nevű szimbolikus paramétert beállítja a newval értékre. Ha a newval egy üres string vagy 0, akkor a szimbolikus paraméter értéke definiálatlanná válik.

#### Visszatérési érték

-1, ha nincs ilyen nevű szimbolikus paraméter,  
0 egyébként.

### 5.1.9. PNL\_itnums()

#### Összefoglalás

PNL_itnums(ptr)	
char *ptr;	a kapott panelpointer

#### Leírás

Visszaadja az aktuális menü tételeinek számát.

#### Visszatérési érték

A menü tételeinek száma.

### 5.1.10. PNL\_name()

#### Összefoglalás

PNL_name(sernum, ptr)	
int sernum;	a tétel indexe
char *ptr;	a kapott panelpointer

#### Leírás

Visszaadja a paraméterek által meghatározott tétel nevét.

#### Visszatérési érték

A paraméterek által meghatározott tétel neve.

### 5.1.11. PNL\_select()

#### Összefoglalás

PNL_select(str)	
char *str;	a kívánt tétel neve

#### Leírás

A rutin meghívásakor ráugrik a panelrendszer megadott nevű tételére, de azt nem hajtja végre. Ezáltal a rutint hívó C függvényben kiválasztható a következőként felajánlott tétel (a tétel nevének megadását l. még "A paneltétel azonosítása a tétel nevével" c. pontban).

#### Visszatérési érték

-1, ha a nincs ilyen nevű tétel a panelrendszerben,  
0 egyébként.

### 5.1.12. PNL\_exec()

#### Összefoglalás

PNL\_exec(str)  
char \*str;

a kívánt tétel neve

#### Leírás

A rutin meghívásakor ráugrik a panelrendszer megadott nevű tételére és végrehajtja azt. Ezáltal a rutint hívó C függvényben kiválasztható a következőként végrehajtandó tétel (a tétel nevének megadását l. még "A paneltétel azonosítása a tétel nevével" c. pontban).

#### Visszatérési érték

-1, ha a nincs ilyen nevű tétel a panelrendszerben,  
0 egyébként.

### 5.2. A futtatórendszer általános függvényei

A futtatórendszer általános függvényei jól használhatók a felhasználó által megírt függvényekben. Az egyes rutinok az angol vagy magyar változattól függően angol vagy magyar üzenetekkel működnek.

### 5.2.1. getanswer()

#### Összefoglalás

```
int getanswer(string)
    char *string;          kiírandó kérdés
```

#### Leírás

Kiírja a megadott stringet egy dinamikus ablakba, hozzáfűzve "? (y/n)" vagy magyar estén a "? (i/n)" stringet és megvárja, amíg értelmes választ kap. Az ESC "No"-nak (ill. "Nem"-nek) minősül. A rutin a képernyőt és a kurzor helyét nem rontja el.

#### Visszatérési érték

1 'Y' és 'y' (ill. 'I' és 'i') esetén,  
0 'N', 'n' és ESC esetén.

### 5.2.2. message()

#### Összefoglalás

```
void message(string)
    char *string;          a kiírandó üzenet
```

#### Leírás

Egy dinamikus ablakba kiírja a megadott stringet és ESC leütéséig várakozik. A rutin a képernyőt és a kurzor helyét nem rontja el.

#### Visszatérési érték

Nincs.

### 5.2.3. myerror()

#### Összefoglalás

```
void myerror(string)
      char *string;           a hibaüzenet
```

#### Leírás

Egy dinamikus ablakba kiírja az "Error!" (ill. "Hiba!") üzenet után a megadott stringet a C runtime megfelelő nyelvű hibaüzenetével együtt (ha standard C függvény értelmezte a hibát). A rutin a képernyőt és a kurzor helyét nem rontja el.

#### Visszatérési érték

Nincs.

### 5.2.4. myderror()

#### Összefoglalás

```
void myderror(string1, string2)
      char *string1;         a hibaüzenet első része
      char *string2;         a hibaüzenet második része
```

#### Leírás

Egy dinamikus ablakba kiírja az "Error!" (ill. "Hiba!") üzenet után a megadott stringeket a C runtime megfelelő nyelvű hibaüzenetével együtt (ha standard C függvény értelmezte a hibát). A rutin a képernyőt és a kurzor helyét nem rontja el.

#### Visszatérési érték

Nincs.

### 5.2.5. myalloc()

#### Összefoglalás

char \*myalloc(size)  
    unsigned size;                      az allokálandó blokk mérete

#### Leírás

Ellenőrzötten foglal le memóriát.

Ha nincs elég rendelkezésre álló terület, akkor az "Out of memory!", illetve a "Kevés a memória!" üzenet kiírása után 255-ös exit kóddal abortáltatja a programot.

#### Visszatérési érték

Az allokált terület kezdőcíme.

### 5.2.6. myrealloc()

#### Összefoglalás

char \*myrealloc(old, size)  
    char \*old;                          a korábban allokált terület  
    unsigned size;                      az allokálandó blokk mérete

#### Leírás

A korábban lefoglalathoz képest több vagy kevesebb memóriát foglal le ellenőrzötten.

Ha nincs elég rendelkezésre álló terület, akkor az

"Out of memory!", illetve a "Kevés a memória!" üzenet kiírása után 255-ös exit kóddal abortáltatja a programot.  
A myrealloc első paramétere lehet 0 is, ekkor a rutin egy myalloc-ot hajt végre.

#### Visszatérési érték

Az újraallokált terület kezdőcíme.

#### 5.2.7. convnum()

##### Összefoglalás

```
char *convnum(num)           a konvertálandó szám
    int num;
```

##### Leírás

Statikus bufferbe konvertálja a megadott számot és visszaadja az első értékes számjegy (vagy a '-' előjel) pointerét a statikus bufferen belül. A statikus buffer neve 'numbuf' és 7 byte hosszúságú (-32768 és a lezáró 0).

A rutin így alkalmas

- egy szám gyors konvertálására,
- a printf "%d" kiírás helyettesítésére és
- a printf "%Nd" kiírás helyettesítésére, ahol  $0 < N < 7$ .

A harmadik funkcióhoz a 'convnum' hívása után a statikus buffert kell kiíratni a megfelelő pozíciótól (numbuf+x, ahol x = 6 - mezőszélesség). Például, ha v = 15, akkor a " 15" kiírást eredményező utasítások:

```
extern char numbuf[];
int v = 15;
.
.
convnum(v);
puttys(numbuf+2);
```

### Visszatérési érték

A statikus bufferben az első számjegy (vagy a '-' előjel) pointerre.

### 5.2.8. scanstr()

#### Összefoglalás

```
char *scanstr(txt,oldstr,maxlen)
char *txt;           a kiírandó prompt
char *oldstr;       az eredeti tartalom
int maxlen;        a maximális hossz
```

#### Leírás

A függvény egy ablakosított stringmegadási lehetőség. A képernyő közepén egy dinamikus ablakba kiírásra kerül a megadott szöveg és az oldstr. Ezután az oldstr módosítható a kurzorbillentyűzet értelemszerű használata mellett. A beütött szöveg egy statikus bufferbe kerül, amely az ENTER billentyűvel való stringlezárás után a függvény visszatérési értéke.

Visszatérési érték

- 1, ha az editálást ESC zárta,
- 0, ha a visszaadott szöveg hossza 0,  
a statikus buffer címe egyébként.

5.2.9. xchgstr()

Összefoglalás

```
char *xchgstr(newstr,oldstr)
    char *newstr;           az új string
    char *oldstr;          a régi string
```

Leírás

Az átadott 'newstr' dinamikus stringre cseréli le az 'oldstr'-et.

A rutin kevés memória esetén a myalloc() mintájára működik.

Visszatérési érték

Az új string címe.

### 5.2.10. newname()

#### Összefoglalás

char \*newname(newstr)  
char \*newstr; az új string

#### Leírás

Ellenőrzött strdup(), tehát dinamikus helyre teszi az átadott stringet.

A rutin kevés memória esetén a myalloc() mintájára működik.

#### Visszatérési érték

Az új string címe.

### 5.3. A futatórendszerbe beépített MSCEX függvények

A futatórendszer nagymértékben támaszkodik az MSCEX run-time könyvtárra. (CEX, (c) Schill Róbert, Magister Software 1987) Ezért a futatórendszer könyvtára a következő MSCEX rutinokat tartalmazza (kódnövekedés nélkül használhatók a felhasználói programban):

agetty(), aputty(), aputtys(), aputtysn(), awrchar(), beep(),  
cupo(), currdisk(), curstype(), dselect(), editline(),  
erase(), frame(), frstfile(), getcpos(), getdir(), getmode(),  
kbflush(), kbget(), movescr(), movframe(), nextfile(),  
parsname(), parspath(), printhlp(), putty(), puttys(),  
qframe(), rdchar(), select(), selfile(), setattr(), setcpos(),  
setmode(), windfram(), windmove(), windrest(), windsave()

#### 5.4. A futtatórendszer változói

char glcolors[5]

a panelrendszer színei, ahol az egyes színek rendre a következők:

- 0 alapszín
- 1 kiemelt szín
- 2 védett szín
- 3 üzenet és hibakiírás színe
- 4 helpkiírási szín

char \*PNL\_root

a panelrendszer gyökere (a főpanel mutatója)

char \*PNL\_finame;

a panelrendszer file-azonosítójának teljes neve.

int PNL\_blackandwhite

a futtatórendszer fekete-fehér módban van-e elindítva.

## 6. A futtatórendszer operátori kezelése

### 6.1. A felhasználói program indítása

A felhasználói program indítása a következő módon történhet (ez vonatkozik a PANEL.EXE és a PANELM.EXE programra is):

```
>[path]program [+b] [+a argfile] [+p panelfile] [[userarg] ]
```

ahol

<path> a programot tartalmazó könyvtár,

<program> a felhasználói .EXE program,

'+b' a fekete-fehér üzemmódot jelöli,

<argfile> a korábbi futtatáskor kimentett szimbolikus paraméterek file-ja (a '+a' után szereplő szóköz opcionális)

az <argfile> könyvtára alapértelmezésben a <program> könyvtára, tehát ha a <program> és az <argfile> azonos könyvtárban vannak, akkor az <argfile> könyvtárát nem kell megadni.

ha az <argfile> és a <program> neve megegyezik (természetesen a file típusán kívül) és azonos könyvtárban is vannak, akkor a <argfile>-t nem kell megadni.

<panelfile> a CREAP által létrehozott panelfile

a '+p' után szereplő szóköz opcionális.

a <panelfile> könyvtára alapértelmezésben a <program> könyvtára, tehát ha a <program> és a <panelfile> azonos könyvtárban vannak, akkor a <panelfile> könyvtárát nem kell megadni,

ha a <panelfile> és a <program> neve megegyezik (természetesen a file típusán kívül) és azonos könyvtárban is vannak, akkor a <panelfile>-t nem kell megadni,

<userarg> a felhasználó programja által kezelendő program-paraméterek.

ezeknek a PANEL.EXE indítása esetén nincs értelme.

## 6.2. A felhasználói program szolgáltatásai

Futtatási fázisban az egyes funkciók a következők:

- a panel indulásakor az alatta lévő képernyőrész lementése és a panel kiírása a képernyőre,
- az ablak mozgatása a lehetőségektől (méret, elhelyezkedés) függően balra, jobbra, felfelé és lefelé a Ctrl+kurzorvezérlő billentyűkkel,
- egy tétel kiválasztása és végrehajtása a kurzorvezérlőkkel és az ENTER billentyű leütésével,
- egy tétel kiválasztása és végrehajtása a tételazonosító karakter megadásával,
- kiszállás az aktuális menüből az ESC billentyűvel,
- segítő információ kiírása az aktuális (kiemelt színnel jelölt) tételről, vagy az egész rendszerről,
- kezelési segédlet kérése az Alt-F1 vagy a ? billentyűvel (itt a futtatási fázisban a generálási fázistól eltérően csak az adott pillanatban érvényes billentyűk magyarázata jelenik meg),
- általános segítő információ kérése a Ctrl-F1 billentyűvel,
- segítő információ kérése az aktuális tételről az F1 billentyűvel,

- DOS parancssorozat által értelmezett szimbolikus paraméterek módosítása, beolvasása file-ból és kiírása file-ba az F2, F3 és F4 billentyűk segítségével,
- globális paneltétel hívása a megfelelő (F5, F6, F7 vagy F8) funkcióbillentyű leütésével,
- kilépés a DOS operációs rendszerbe az F9 billentyű segítségével,
- direkt ráúgrás a panelrendszer bármely létező tételére az F10 billentyű leütésével és a kívánt tétel (vagy alpanel) nevének megadásával (l. még "A paneltétel azonosítása a tétel nevével" c. pontban).
- a panelből való kiszálláskor az eredeti képernyőtartalom visszamentése.

## 7. Példa a CREAP futtatórendszer használatára

### 7.1. A CENV program specifikációja

Példaprogramunk egy úgynevezett programozási környezet vezérlő-programja, amely a Microsoft C környezethez csatlakozik.

A példa egy működő angol nyelvű szoftver része. A hitelesség érdekében a rendszert nem magyarosítottuk. Ettől függetlenül a CREAP környezet használatát kellőképpen demonstrálja az angolul nem tudók számára is.

A példaprogram természetesen csak magát a monitort tartalmazza, a környezet által hivatkozott rendszerek (Documentation, Utilities) nem tartoznak a CREAP környezetéhez.

Célunk egy olyan monitor létrehozása volt, amelyből a programfejlesztés során nem kell kilépni, keretet ad a fejlesztendő C program teljes életciklusához.

A monitor által vezérlendő főbb elemek:

- editálás
- fordítás
- szerkesztés
- futtatás
- archiválás
- dokumentálás
- segédprogramok hívása

## 7.2. A CENV panelrendszerének definiálása

A panelrendszer definiálása a CREAP program segítségével körülbelül másfél órai munkát igényel, beleértve a HELP rendszer megírását is.

Jelen pontban csupán a definiálás egyes részeit emeljük ki, illetve magyarázzuk részletesebben, hiszen a disztributív lemezen található CENV.PNL file alapján a létrehozott panelrendszer jól tanulmányozható.

A főpanelünk a következő:

```
-----  
| C Programming Environment |  
-----  
| Edit                      |  
| Icompile & link          |  
| IMake                     |  
| ICompiler                 |  
| ILink                     |  
| IeXecute                  |  
| IDocumentation          |  
| IArchiving                |  
| IUtilities                |  
| INew screen               |  
-----
```

A főmenüben szereplő nagybetűk a kiválasztó karaktereket jelölik. A panel tételei közül (következő szintű) alpanel a "Documentation" és a "Utilities" tétel, C végpont a "New screen" és a "Utilities" egyes tételei, a többi végpont DOS parancssorozatokból áll.

Összetettsége miatt itt a "compile & link" végpont parancssorozatát ismertetjük:

1. msc \$(CFLAGS) \$(ACTFILE);
2. @ifn 1 goto 6
3. @pause "\$(ACTFILE): hiba!"
4. @d:ne \$(ACTFILE).c
5. @goto 1
6. LINK /NOI \$(ACTFILE);
7. @if 1 pause "\$(ACTFILE): Link hiba!"
8. @if 1 goto edit
9. @del \$(ACTFILE).map
10. @del \$(ACTFILE).OBJ
11. @goto execute

Szimbolikus Batch paraméterek:

ACTFILE: az aktuális program (forrás, tárgykód stb.) file-jának neve, típus nélkül.

CFLAGS: a Microsoft C compilernek átadandó opciók (például /Zi/AH)

Az egyes DOS parancsok:

- 1 a Microsoft C compiler hívása
- 2 hibátlan fordítás esetén ugrás a 6. utasításra
- 3 hibás fordítás, hibaüzenet kiírása, majd
- 4 a forrás editálása javításhoz, és
- 5 ugrás az újabb fordításra.
- 6 szerkesztés
- 7 hiba esetén üzenet és
- 8 ráállás a főmenü editálás (Edit) tételére.

- 9 a szerkesztési napló törlése
- 10 a tárgykód törlése
- 11 a főmenü végrehajtási tételére (execute) való ráállítás.

A "Utilities" alpanel segítségével néhány hasznos segédprogram elindítását tesszük lehetővé. A segédprogramok a TOP40 programcsomag elemei (TOP40, (c) Schill Róbert, Magister Software, 1988)

Utilities
lAlarm
lscopy
lDel
lcmPdir
ldf
lfMove
lGrep
lUniq

A panel minden egyes végpontja az "exeutil" nevű C függvény. Az "exeutil" függvény (l. a forrásprogramnál) paraméterei segítségével határozza meg a meghívandó segédprogramot, lehetőséget ad a segédprogram interaktív paraméterezésére, majd elindítja a segédprogramot.

A globális paneltételekhez (F5, F6, F7 és F8) gyakran használt funkciókat rendelünk, amelyeket C rutinokkal valósítunk meg (a rutinok forrását l. a következő pontban):

- F5: könyvtár tartalomjegyzék kérése  
C függvény: "dir()"
- F6: file törlése  
C függvény: "delete()"

F7: file listázása (LI.COM hívása)  
C függvény: "list()"

F8: file editálása (NE.COM hívása, Norton Editor)  
C függvény: "edit()"

### 7.3. A CENV-ről a CREAP által készített dokumentáció

Az alábbiakban a CENV panelrendszerrel generált dokumentáció látható. A dokumentáció teljesen automatikusan képződik generálási fázisban az Alt-D billentyű hatására.

A dokumentáció két fő részből áll:

- összefoglaló táblázat a panelrendszer minden egyes tételéről, egy sorban (táblázat forma),
- részletes dokumentáció minden egyes tételről.

Name	Typ	Hlp	reference to
C Programming Environment	Panel	+	
Edit	DOS	+	d:ne \$(ACTFILE).C
compile & link	DOS	+	msc \$(CFLAGS) \$(ACTFILE);
Make	DOS	+	make \$(NAME).EXE
Compiler	DOS	+	msc \$(CFLAGS) \$(ACTFILE);
Link	DOS	+	link \$(LINKF) @\$\$(NAME).LNK;
eXecute	DOS	+	\$(NAME) \$(EXECARGS)
Documentation	Panel	+	
structure Diagram	DOS	+	csd \$(CSDF) -f \$(NAME).SUM
Procedure documenter	DOS	+	cpd \$(CPDF) -f \$(NAME).SUM
Symbol crossref	DOS	+	cdx \$(CDXF) -f \$(NAME).SUM
Module symbol crossref	DOS	+	cmx \$(CMXF) \$(ACTFILE)
Include structure	DOS	+	cinc \$(CINCF) -f \$(NAME).SUM
style Analys	DOS	+	csa \$(CSAF) -f \$(NAME).SUM
Archiving	DOS	+	del *.*
Utilities	Panel	+	
Alarm	C funct	+	exeutil()
sCopy	C funct	+	exeutil()
sDel	C funct	+	exeutil()
cmPdir	C funct	+	exeutil()
dF	C funct	+	exeutil()
fMove	C funct	+	exeutil()
Grep	C funct	+	exeutil()
Uniq	C funct	+	exeutil()
New screen	C funct	+	erase()
----- Global F5 -----			
dir	C funct	-	dir()
----- Global F6 -----			
delete	C funct	-	delete()
----- Global F7 -----			
list	C funct	-	list()
----- Global F8 -----			
edit	C funct	-	edit()

```
| C Programming Environment |
|-----|
|Edit                       |
|compile & link             |
|Make                       |
|Compiler                   |
|Link                       |
|eXecute                   |
|Documentation              |
|Archiving                  |
|Utilities                  |
|New screen                 |
|-----|
```

Select letters: "E&MCLXDAUN"

Help text:

for Microsoft C (4.00 and above)

Version 1.00

Produced by Robert Schill  
1987.jun.

---

The CENV is an integrated system for C programmers to develop their programs. The system helps the user in

- editing
- compiling and linking
- testing
- executing
- documenting
- archiving
- file and directory handling

This is the userfriend monitor of the CENV system. Usage of the monitor is very simple. For help press the '?' key. For more detail see the description.

```
-----  
| Edit |  
-----
```

Calls the DOS commands:

```
d:ne $(ACTFILE).C  
@goto compile & link
```

Help text:

The EDIT function calls the NORTON Editor with the actual value of the "ACTFILE.c" (symbolic batch argument).

```
-----  
| compile & link |  
-----
```

Calls the DOS commands:

```
msc $(CFLAGS) $(ACTFILE);  
@ifn 1 goto 6  
@pause "$(ACTFILE): hiba!"  
@d:ne $(ACTFILE).c  
@goto 1  
LINK /NOI $(ACTFILE);  
@if 1 pause "$(ACTFILE): Link hiba!"  
@if 1 goto edit  
@del $(ACTFILE).map  
@del $(ACTFILE).OBJ  
@goto execute
```

Help text:

First compiles (Microsoft) the "ACTFILE" then links it.  
If an error detected during compiling calls the EDIT panel.

Make

Calls the DOS commands:  
make \$(NAME).EXE

Help text:  
Starts the 'make' utility with  
the target "NAME.exe".

Compiler

Calls the DOS commands:  
msc \$(CFLAGS) \$(ACTFILE);  
@ifn 1 goto 5  
@ask "\$(ACTFILE): hiba! Editing" goto 7  
exit  
@ask Link pntexec link  
exit  
d:ne \$(ACTFILE).c  
goto 1

Help text:  
Calls the Microsoft C compiler  
with the argument "ACTFILE.c".

Link

Calls the DOS commands:  
link \$(LINKF) @\$(NAME).LNK;  
goto execute

Help text:

Calls the Microsoft C Linker  
with the argument "NAME.lnk"  
linker command-file.

-----  
execute

Calls the DOS commands:  
\$(NAME) \$(EXECARGS)

Help text:

Executes the "NAME.exe" program  
with the USERARG program  
arguments.

-----  
Documentation
Structure Diagram
Procedure documenter
Symbol crossref
Module symbol crossref
Include structure
Style Analys
-----

Select letters: "DPSMIA"

Help text:

This is the panel of the MI Software  
C documentation package.

The facilites of the package:

Structure diagramm  
with alphabetic content

of procedures  
Procedure document  
with the graphic  
tree of procedure calls  
Documentation of symbols  
Include structure  
crossreference,  
cross table  
makefile dependencies  
generation  
Style analysis about  
identification  
rate of comments  
length of symbols

-----  
structure Diagram

Calls the DOS commands:  
csd \$(CSDF) -f \$(NAME).SUM  
@goto procedure doc

Help text:

Not yet implemented.

-----  
Procedure documenter

Calls the DOS commands:  
cpd \$(CPDF) -f \$(NAME).SUM  
@goto symbol cross

Help text:

Not yet implemented.

-----  
Symbol crossref

Calls the DOS commands:

```
cdx $(CDXF) -f $(NAME).SUM  
@goto module symbol
```

Help text:

Not yet implemented.

-----  
Module symbol crossref

Calls the DOS commands:

```
cmx $(CMXF) $(ACTFILE)  
@goto Include str
```

Help text:

Not yet implemented.

-----  
Include structure

Calls the DOS commands:

```
cinc $(CINCF) -f $(NAME).SUM  
goto style anal
```

Help text:

Not yet implemented.

| style Analys |

Calls the DOS commands:

```
csa $(CSAF) -f $(NAME).SUM
@goto archiving
```

Help text:

Not yet implemented.

| Archiving |

Calls the DOS commands:

```
del *. *
ds en >null
save $(ACTDIR) $(SAVEVOL)
```

Help text:

- Deletes the backup files of the NORTON Editor
- Sorts the directory by extension and names of the files
- Calls the 'save' utility saving into the ACTDIR directory on diskette SAVEVOL.

Utilities	
lAlarm	
lscopy	
lDel	
lcmPdir	
ldf	
lmove	
lgrep	
luniq	

Select letters: "ACDPFMGU"

Help text:

Opens a window to a usefull set of TOP40 Utilities. The TOP40 Utilities displays their usage if -h option is given.

Alarm
Alarm

Calls C function "exeutil()" with sernum: 0

Help text:

Installes (or deletes) an alarm time and message.

scopy
scopy

Calls C function "exeutil()" with sernum: 1

Help text:

Selected copy from a path  
to an other path.

| sDel |

Calls C function "exeutil()" with sernum: 2

Help text:

Selected delete of files.

| cmPdir |

Calls C function "exeutil()" with sernum: 3

Help text:

Compares the directories  
and updates both.  
Prompts for copying  
the nonexistent files.

| dF |

Calls C function "exeutil()" with sernum: 4

Help text:

Displays on the screen the  
sorted filenames, the sum-  
ma size of them and the  
free spece on the disk.  
Specifiing (-c) the start  
column, more directories

can displayed on the same screen.

`fMove`

Calls C function "exeutil()" with sernum: 5

Help text:

Moves files from a directory to another without physical copying. Modes:

- no overwrite existing files
- overwrite existing files
- selected move

`Grep`

Calls C function "exeutil()" with sernum: 6

Help text:

String search utility. Full compatible with the Unix GREP.

`Uniq`

Calls C function "exeutil()" with sernum: 7

Help text:

Searches the files with the same name in the directories of the disk and lets the user to delete the unused copies.

`| New screen |`

Calls C function "erase()" with sernum: 9

Help text:

Clears the screen and positions to home under the panel window.

----- Global F5 -----

`| dir |`

Calls C function "dir()" with sernum: -1

----- Global F6 -----

`| delete |`

Calls C function "delete()" with sernum: -1

----- Global F7 -----

| list |

Calls C function "list()" with sernum: -1

----- Global F8 -----

| edit |

Calls C function "edit()" with sernum: -1

#### 7.4. A CENV.C keretprogram forrása

```
/* CENV.C
**
** source of CENV.EXE
** (C Programming Environment - sample program)
**
** Robert Schill Jun/03/1987
**/

/* Microsoft headers */

#include <malloc.h>
#include <string.h>
#include <process.h>

/* MSCEX declarations */

#define E_ESC ((char *)-1)
extern char *editline();

extern int aputtys();
extern int setattr();
extern int erase();
extern int awrchar();
extern int kbget();
extern int currdisk();
extern char *parsname();

extern int frstfile();
extern int nextfile();

/* File attributes */

#define R_ONLY 0x01
#define HIDDEN 0x02
#define SYSTEM 0x04
#define VOLABEL 0x08
#define SUBDIR 0x10
#define ARCHIVE 0x20
```

```
/* structure of Disk Transfer Area */
```

```
struct FC {  
    char        f_res[21];  
    char        f_attr;  
    unsigned    f_time;  
    unsigned    f_date;  
    long        f_size;  
    char        f_nam[13];  
};
```

```
/* macros for decode f_date and f_time */
```

```
#define F_HOUR(t)      ((t >> 11)  
#define F_MIN(t)       ((t >> 5) & 0x3F)  
#define F_SEC(t)       ((t & 0x1F) << 1)  
  
#define F_YEAR(t)      ((t >> 9) + 1980)  
#define F_MONTH(t)     ((t >> 5) & 0xF)  
#define F_DAY(t)       (t & 0x1F)
```

```
#ifndef NULL
```

```
    #define NULL    0
```

```
#endif
```

```
#define YES    1
```

```
#define NO     0
```

```
#define MAXTEL 16
```

```
main(argc,argv)
```

```
int argc;
```

```
char *argv[];
```

```
{
```

```
    inicfuncs();           /* connect C function table (by CREAP) */
```

```
    inipanel(argc,argv);  /* process arguments, get panel & args */
```

```
    dopanel();            /* execute panel structure */
```

```
    exit(0);
```

```
}
```

```
/* C panel endpoints referenced in CENV.PNL */
```

```
int exeutil(sernum,ptr)
/* Execute the actual utility */
int sernum;
char *ptr;
{
char *PNL_name();
char *actexec;
register char *p;

actexec = PNL_name(sernum,ptr);
aputtys(actexec,7);
aputtys(" ",7);
p = editline(NULL,79,80,31,112,'\r');
aputtys("\r",7);
setattr(80,7);
aputtys("\r\n",7);
if (p == _E_ESC)
    return;

if (spawnlp(P_WAIT,actexec,actexec,p,NULL) < 0)
    myerror(actexec);
}

char *months[13] =
    {"","Jan","Feb","Mar","Apr",
     "May","Jun","Jul","Aug",
     "Sep","Oct","Nov","Dec"};

char wildcards[] = "*.*";

dir()
{
struct FC fc;
register struct FC *pfc;
register unsigned w;
char *p;

erase();
aputtys("Dir ",7);
```

```
p = editline(wildcards,79,80,15,112,'\r');
if (p == NULL || p == E_ESC)
    return;
pfc = &fc;
if (frstfile(p,pfc,ARCHIVE!SYSTEM!R_ONLY!HIDDEN) == 0)
{
    myderror(p,": no such file");
    return;
}
do
{
    w = pfc->f_date;
    printf("\n%-15s %03o %s/%02d/%4d",
    pfc->f_nam,pfc->f_attr,months[F_MONTH(w)],F_DAY(w),F_YEAR(w));
    w = pfc->f_time;
    printf(" %02d:%02d %7ld bytes",
    F_HOUR(w),F_MIN(w),pfc->f_size);
}
while(nextfile(pfc));
anykey();
}

delete()
{
    struct FC fc;
    register struct FC *pfc;
    char buffer[80];
    char *pb;
    char *p;

    erase();
    aputtys("Delete ",7);
    p = editline(wildcards,79,80,15,112,'\r');
    if (p == NULL || p == E_ESC)
        return;
    if ( (pb = parsname(currdisk()),p,buffer)) == NULL)
    {
        myerror("Bad path");
        return;
    }
}
```

```
awrchar(73, ' ', 7);
aputtys(buffer, 15);
aputtys("?", 15);
if (toupper(kbget() & 0xFF) != 'Y')
{
    aputtys("No", 7);
    return;
}
aputtys("Yes", 7);
pfc = &fc;
if (frstfile(buffer, pfc, ARCHIVE) == 0)
{
    myderror(p, ": no such file");
    return;
}
do
{
    strcpy(pb, pfc->f_nam);
    printf("\n%-15s %s deleted",
        buffer, unlink(buffer) == -1 ? "NOT:");
}
while(nextfile(pfc));
anykey();
}

list()
{
    static char listcom[] = "li.com";
    char buffer[80];
    char *p;

    erase();
    aputtys("List ", 7);
    p = editline(wildcards, 79, 80, 15, 112, '\r');
    if (p == NULL || p == _EESC)
        return;
    if (spawnlp(P_WAIT, listcom, listcom, p, NULL) < 0)
        myerror(listcom);
}
}
```

```
edit()
{
static char necom[] = "ne.com";
char buffer[80];
char *p;

erase();
aputtys("Edit ",7);
if ((p = editline(NULL,79,80,15,112,'\r')) == NULL || p == E_ESC)
return;
if (spawnlp(P_WAIT,necom,necom,p,NULL) < 0)
myerror(necom);
}

anykey()
{
aputtys("\r\nAny key...",7);
kbget();
aputtys("\r\n",7);
}

#include "CENV.CP"

/* end */
```

### 7.5. A CENV.CP generált forrásfile

```
struct PNL_Func {
    int    (*userfunc)();
    int    screened;
};

extern int exeutil();
extern int dir();
extern int delete();
extern int list();
extern int edit();
extern int erase();

static struct PNL_Func localfuncs[6] = {
    {exeutil,1},
    {dir,1},
    {delete,1},
    {list,1},
    {edit,1},
    {erase,1}
};

inicfuncs()
{
    extern struct PNL_Func *PNL_funcs;
    PNL_funcs = localfuncs;
}
```

## 7.6. A CENV.EXE generálása

A végrehajtható CENV monitor a következő DOS paranccsal generálható:

```
>CL cenv.c -link panel
```

## 7.7. A CENV.EXE használata

A CENV indítása a

```
>CENV
```

paranccsal lehetséges.

Fekete-fehér képernyő esetén:

```
>CENV +b
```

Indításkor a CENV.EXE monitor automatikusan betölti a CENV.PNL panelfile-t és a CENV.ARG szimbolikus paraméterek file-ját. Indítás után a szimbolikus paramétereket az F2 billentyű leütése után lehet módosítani (első indítás esetén definiálni). A módosított szimbolikus paramétereket az F4 billentyű segítségével írhatjuk ki file-ba.

A CENV használata innen kezdve értelemszerű.

Példa:

- 1 Indítsuk el a CENV-et a >CENV parancs segítségével.
- 2 Üssük le az F2 billentyűt a szimbolikus paraméterek módosításához.
- 3 Allítsuk be az ACTFILE és a NAME szimbolikus paramétereket a "PROBA" értékre.

- 4 Az ESC billentyűvel térjünk vissza a főmenühez.
- 5 Az "e" betű leütésével hozzuk létre és editáljuk meg a PROBA.C file-t.
- 6 Editálás után az aktuális tétel a főmenü "compile & link" tétele, üssük le az ENTER billentyűt.
- 7 Innen a korábbiakban részletezett DOS parancssorozatot hajtódik végre.
- 8 Sikeres szerkesztés után az aktuális tétel az "execute".
- 9 Az ENTER billentyű segítségével a létrehozott "PROBA.EXE" programunkat elindíthatjuk. Ha ez programparamétereket is használ, akkor indítás előtt az F2 billentyű leütése után állítsuk be az EXECARGS szimbolikus paramétert a kívánt értékre.
- 10 Programunk lefutása után üssük le az F10 billentyűt.
- 11 A megjelenő ablakba írjuk be, hogy "sdel", ezzel direkt módon ráugorhatunk a "Utilities" alpanel "sDel" tételére.
- 12 Az SDEL segédprogram számára adjuk meg a "PROBA.\*" paramétert.
- 13 ENTER után a PROBA nevű file-ok közül eltörölhetjük a fölöslegeseket.
- 14 ESC hatására visszatérünk a főmenühez.
- 15 Újabb ESC hatására megjelenik az "Are You sure to return to DOS?" (Biztos, hogy vissza akar térni a DOS-ba?) kérdés.
- 16 "y" válasszal kiszállhatunk a CENV rendszerből.

## 8. A disztributív lemez tartalma

A disztributív lemezen a következő állományok találhatóak:

CREAP.EXE a panelrendszert létrehozó és karbantartó program;

PANEL.EXE standard keretprogram, csak DOS parancssorozatot tartalmazó panelrendszerhez;

PANELM.EXE a PANEL.EXE magyar nyelvű üzenetekkel működő változata;

PANEL.LIB a futtatórendszert tartalmazó tárgykódkönyvtár;

PANELM.LIB a PANEL.LIB magyar nyelvű üzenetekkel működő változata;

CENV.PNL a példaprogram panelfile-ja;

CENV.C a példaprogram C nyelvű forrása;

CENV.HLP a példaprogram helpfile-ja;

CENV.EXE a végrehajtható példaprogram;

SK.PNL panelfile, amely a Pull-down menüt mutatja be (a Borland cég SuperKey rendszerének szimulálása);

SK.C keretprogram az SK.PNL használatához;

SK.EXE a végrehajtható Pull-down példaprogram.

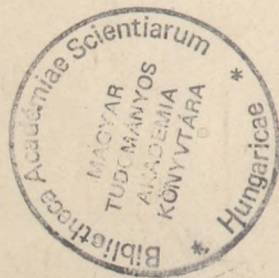


## Index

- Az ablakok helyének meghatározása 36
- Az ablakok méretének módosítása 37
- Az ablakok paramétereinek meghatározása 36
- Az ablakok színeinek meghatározása 38
- Az aktív panel befejezőfüggvénye 50
- Az aktív panel kezdőfüggvénye 49
  - Aktív panelek definiálása 19
  - Alpanelek kimentése és beolvasása 17
  - ASK 31
  - BEEP 32
  - Befejező C függvény meghatározása 20
  - Bevezetés 2
- A billentyűzet használata 9
  - C függvényt hívó végpont 26
- A CENV panelrendszerének definiálása 75
- A CENV program specifikációja 74
- A CENV-ről a CREAP által készített dokumentáció 78
- A CENV.C keretprogram forrása 92
- A CENV.CP generált forrásfile 98
- A CENV.EXE generálása 99
- A CENV.EXE használata 99
  - convnum() 66
- A CREAP által kezelt file-ok 7
- A CREAP rendszer indítása 6
- A CREAP verzió kiírása 44
- A disztributív lemez tartalma 102
  - dopanel() 56
  - DOS parancssorozatot hívó végpont 28
- Az elszámolást végző rutinok 51
  - Elszámoló rutinok beépítése 21
  - EXIT 32
- A felhasználó által használható függvények és változók 55
- A felhasználói C függvények 49
- A felhasználói program indítása 71
- A felhasználói program szolgáltatásai 72
- A futtatórendszer általános függvényei 62
- A futtatórendszer használata 46
- A futtatórendszer operátori kezelése 72
- A futtatórendszer panelkezelő függvényei 54

- A futtatórendszer változói 70
- A futtatórendszerbe beépített MSCEX függvények 69
  - getanswer() 63
  - Globális alpanelek 18
  - GOTO 32
- A helpfile illesztése a panelrendszerhez 41
  - IF 33
  - IFN 34
  - inipanel() 55
- A keret típusának állítása 39
  - Kezdő C függvény meghatározása 20
  - kezelési segédlet megjelenítése 9, 12, 43
  - Kilépés a DOS-ba 44
  - Kiszállás a rendszerből, output file-ok képzése 8
- A kiterjesztett DOS parancsnyelv 31
  - message() 63
  - myalloc() 65
  - myderror() 64
  - myerror() 64
  - myrealloc() 65
  - newname() 69
  - Panelek átkötése, ismétlése 22
- A panelrendszer definiálása, karbantartása 5
- A panelszerkezet megjelenítése és dokumentálása 43
- A paneltétel azonosítása a tétel nevével 13
- A paneltétel törlése 16
- A paneltételek definiálása 15
  - Paneltételek sorrendjének megváltoztatása (felcserélés) 25
- A paneltételeket azonosító karakterek meghatározása 40
  - PAUSE 34
  - Példa a CREAP futtatórendszer használatára 75
  - PNLEXEC 35
  - PNL\_exec() 62
  - PNL\_getarg() 59
  - PNL\_glisprotect() 59
  - PNL\_glprotect() 58
  - PNL\_isprotect() 57
  - PNL\_itnums() 60
  - PNL\_name() 61
  - PNL\_protect() 56
  - PNL\_select() 61

PNL_setarg()	60
Pull-down menük	40
scanstr()	67
A szimbolikus paraméterek	29
Tétel típusának megváltoztatása	25
A tételek nevének módosítása	39
Új paneltétel beszúrása	16
Végpontok meghatározása	26
A végpontot lefedő C függvény	48
xchgstr()	68



AKADÉMIAI KIADÓ ÉS NYOMDA VÁLLALAT

Felelős kiadó: Hazai György főigazgató

Felelős szerkesztő: Votisky Zsuzsa

A borítót Szabadi Zoltán tervezte

ISBN 963 05 4872 0

Kézirat lezárva: 1988. február

OPTIMUM – MGKSZ Ny. 88.073





## CREAP - mint CREAtE Panel -

hierarchikus menü ill. panelgeneráló és futtató rendszert.

Az IBM PC/XT/AT és a vele kompatibilis számítógépek korszerű szoftverei és felhasználói programjai hierarchikus, „ablakos” menürendszerrel fordulnak a felhasználó felé, hogy a szolgáltatások szövevényében tájékozódást nyújtsanak.

A CREAP segítségével ez a legkorszerűbb technika könnyedén megvalósítható. A menüpontokhoz jól paraméterezhető DOS parancssorozatok vagy C függvények köthetők.

A program a panelépítés minden részletét ellátja: az ablakok definiálásától a menüpontok hierarchiájának és tartalmának meghatározásán át a nyomtatható dokumentációig.

Két fázisa: a generálórendszer és a futtató rendszer.

Demonstrációként egy önmagában is jól használható C programozói környezetet tartalmaz.

Felhasználható:

- PC-DOS/MS-DOS 3.xx operációs rendszerre épülő menürendszer kialakítására.
- MICROSOFT C 4.00 vagy 5.00 környezetben készülő professzionális monitor létrehozására.
- Még kényelmesebb C környezetet biztosít a CEX kiterjesztett C könyvtár együttes alkalmazásával.

Hardver igény: IBM PC/XT/AT ill. kompatibilis mikroszámítógép.

Garanciális szolgáltatások,

szoftverkövetés,

igény szerinti betanítás.

