

DOKUMENTUMOK


(D-5)

UNI 2.0

User Network Interface

Felhasználói kézikönyv

Budapest, 1988. november



Készült az Információs Infrastruktúra Fejlesztési Program
támogatásával az MTA SZTAKI IBM Szoftver Osztályán

A kiadásért felel: Bakonyi Péter

Írta: Szentjóni Ottó

A rendszert készítették:

Böjthe Zoltán
Szentjóni Ottó
Terge Zsuzsa

A hálózati illesztőt az MTA SZTAKI Hálózati Főosztályán
fejlesztették.

TARTALOM

1. Bevezetés	1
1.1. Az UNI program főbb jellemzői	1
1.2. Az ékezetes karakterek kezelése	3
2. Az UNI program kezelése	4
2.1. A képernyő	7
2.2. Az UNI program parancsállapotából kiadható parancsok	9
3. Az UNI parancsfile nyelv leírása	11
3.1. Stringkonstansok	12
3.2. A vevőbuffer	13
3.2. Az UNI parancsnyelv utasításai	14
3.3. Példaprogram	24
4. Az UNI installálása	27
4.1. A disztribúciós lemez tartalma	28
Mutató	30

Magyar Tudományos Akadémia Kísérleti Fizikai Intézetének Közleményei 1972. évi 1. kötet

TARTALOM

1	1. Bevezetés
1	1.1. Az UNI program főbb jellemzői
2	1.2. Az érzékelés hardverének kialakítása
4	2. Az UNI program kialakítása
5	2.1. A képernyő
7	2.2. Az UNI program parancsleírása
8	Képernyő parancsok
11	3. Az UNI parancsleírás nyelv leírása
12	3.1. Sztringkonstrukciók
12	3.2. A vezérlés
14	3.3. Az UNI parancsleírás nyelvének leírása
24	3.4. Példaprogram
27	4. Az UNI tesztelés
28	4.1. A tesztelés leírása
30	Mutató

Hozott anyagról sokszorosítva

1. Bevezetés

A nagygépes szolgáltatások (pl adatbázisok) hálózati elérése PC-ről a jelenleg rendelkezésre álló szoftvereszközökkel (KERMIT, PROCOMM, stb) rendkívül nehézkes: kézzel kell felhívni a postai X25 hálózatot, ott esetleg PAD paramétereket beállítani, tudni kell a host gép hálózati címét (pl. 2801015020), stb. Megoldatlan ezenkívül a magyar ékezetes karakterek (esetleg szolgáltatásokként és munkaállomásokként eltérő) kezelése, továbbítása is.

A User-Network Interface program (UNI) lehetővé teszi, hogy a hálózati hívást és a vonal kezelését automatikusan végeztessük el a PC-vel. A program az IIF keretrendszerbe ágyazható, amely menüvezérelt módon biztosítja az egyes szolgáltatások elérését. Így a szolgáltatást igénybevevő személynek (végfelhasználónak) csak a megfelelő menüelemre kell pozicionálnia a kurzort, és a rendszer automatikusan "bejelentkezteti" őt a távoli szolgáltatásba a hálózaton keresztül. A felhívási, bejelentkezési eljárások, az ékezetes karakterek kezelése, stb. szöveges file-okban írható le.

Jelen dokumentáció az UNI program 2.0 verziójának felhasználói leírását tartalmazza.

1.1 Az UNI program főbb jellemzői

Az UNI egy beépített programozási nyelvet tartalmaz, amelyen hálózati hívási protokollok írhatók le, és képes ezen a nyelven írt programok végrehajtására. Egy ilyen program futásának eredményeként kerülhetünk kapcsolatba egy távoli szolgáltató számítógéppel. Mivel a hálózat heterogén, a szolgáltatások köre állan-

dóan bővül és a helyi körülményeket is figyelembe kell venni, a parancsfile-okat nem lehet mindenki számára egységesen, "közpon-tilag" elkészíteni, azokat egyedileg kell megírni. Ezért az UNI installálása minden esetben minimális programozói munkát igényel. A programokat tartalmazó szöveges parancsfile-okat az UNICOMP segédprogrammal le kell fordítani, majd az így kapott parancsfile az UNI program által közvetlenül értelmezhető és végrehajtható lesz. Egy tipikus parancsfile a következő részekből állhat:

- A szolgáltató host felhívása (hálózatfüggő)
- Bejelentkezés a szolgáltatásba (LOGON, DIAL, a szolgáltatás behívása, user-azonosítás stb.). Ez a rész hálózatfüggetlen, de felhasználófüggő, hiszen mindenkinek más pl. a felhasználói azonosítója, jelszava stb.
- Interaktív rész (teletype terminál-emuláció). Itt opcionális és programozható karakterfordítás történik mind adási, mind vételi irányban. Így oldható meg a szolgáltatásonként eltérő ábrázolású magyar ékezetes karakterek lefordítása a munkaállomásonként eltérő reprezentációra. Az ékezetes fordítás általános esetben karaktersorozat -> karaktersorozat fordítást jelent, természetesen a karaktersorozat egy karakterből is állhat. A végfelhasználó (a szolgáltatás igénybevevője) csak a teletype-emulációval találkozhat, a többi rész "el van takarva" előle.
- Kijelentkezés a szolgáltatásból (hálózatfüggetlen)
- A kapcsolat lebontása (hálózatfüggő)

Az UNI program rendelkezik néhány kiegészítő funkcióval is (az érkezett karakterek visszalapozása a képernyőn, naplófile készítése stb.).

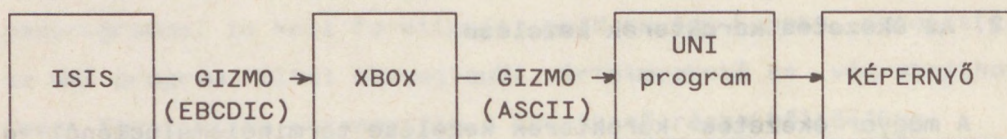
1.2. Az ékezetes karakterek kezelése

A magyar ékezetes karakterek kezelése terminálemulációnál a következőképpen történik:

Mind adási, mind vételi irányban a karakterfordítás egy-egy külön szöveges file-ban írható le, amely parancsfile-ból betölthető és aktivizálható. Általános esetben max. 4 hosszú karakter-sorozatok max. 4 hosszú karaktersorozatokká való fordításáról van szó. Az ISIS adatbáziskezelő rendszerhez adott a nemzeti karakterek GIZMO kódtáblája, amelyben egy nemzeti karaktert egy három ASCII karakterből álló sorozat ábrázol. Ez egy escape-karakterből ("@"), ékezetkódból és karakterkódból áll. Így pl az "ö" betűt GIZMO-kódolásban a "@EO" sorozat ábrázolja. A programhoz mellékelt GIZMO.IN ill. GIZMO.OUT file-ok a GIZMO kódok fordítását írják le egy olyan IBM PC karakterreprezentációra, ahol a magyar ékezetes karakterekhez hasonló karakterek találhatóak. Ha a távoli szolgáltató gép GIZMO-tól eltérő ékezetes karakterreprezentációt használ, vagy a lokális munkaállomáson kívánjuk máshová definiálni az ékezetes karaktereket (pl. mert ROM cserét hajtottunk végre, vagy cirill betűs adatbázisokat kérdezzük le), egy-egy megfelelő fordító file-t kell létrehoznunk mind adási, mind vételi irányban és az UNI számára érvényesítenünk kell (l. "rectrans" ill. "sendtrans" utasítások!). Az UNI program ezek után a file-ban leírt minden karaktersorozatot a neki megfelelő karaktersorozatként ír ki a képernyőre (vételi oldal) ill. minden begépelte karaktersorozatot a megfelelő karaktersorozatként

UNI 2.0

továbbít (adási oldal). A lemezen természetesen tetszőleges számú fordító file lehet, de ezek közül egyszerre csak egy-egy lehet aktív.



A hálózat saját fordítása

Az UNI fordítása

2. Az UNI program kezelése

Az ékezetes fordításon kívül az UNI program jellemzője, hogy a TERMULATOR és a PROCOMM programokhoz hasonlóan képes előre megírt hívási protokollokat végrehajtani. A protokollok az UNI saját programozási nyelvén írhatók le tetszőleges szövegszerkesztővel. A parancsfile minden esetben .CMD kiterjesztésű. A megírt parancsfile-okat az UNICOMP programmal le kell fordítani, amely az esetleges szintaktikus hibákat kiírja a képernyőre és egy <filenév>.ERR nevű hibafailé-ba, illetve szintaktikusan hibátlan parancsfile-ból egy .UNI kiterjesztésű, az UNI program által közvetlenül végrehajtható parancsfile-t készít. Az UNICOMP programot az UNICOMP <szöveges parancsfile-név> [/LINECNT] parancssorral kell indítani. A filenévnek nem kell a .CMD kiterjesztést tartalmaznia, mert ez úgyis alapértelmezés. A /LINECNT opcióval hosszabb lefordított kódot állíthatunk elő, de az ilyen kód nyomkövetése során több információt kapunk (1. SET TRACE parancs ill. utasítás).

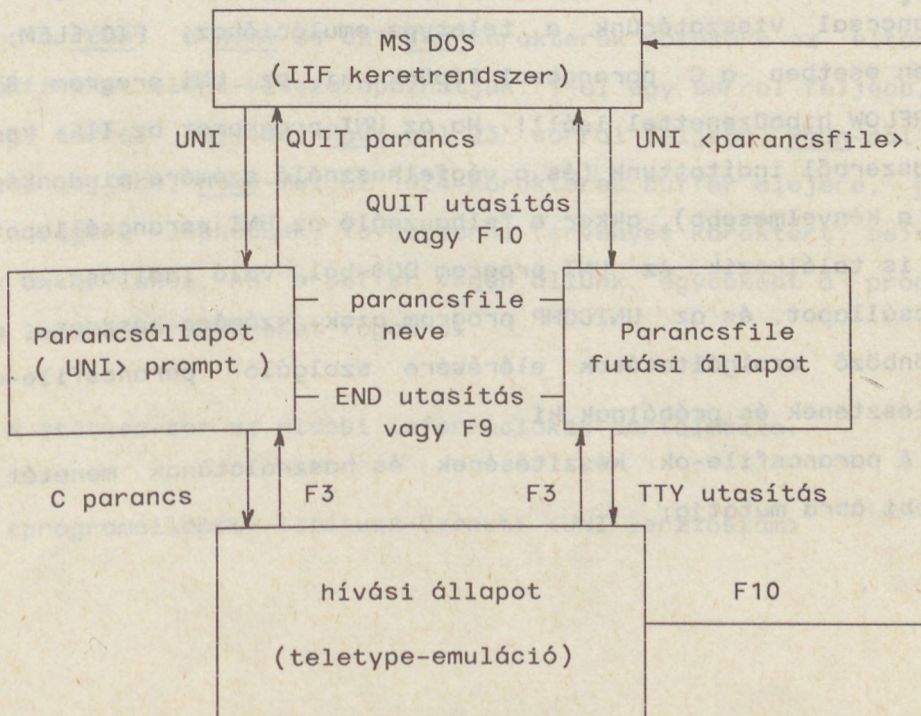
Maga az UNI program DOS-ból az

UNI [<parancsfile-név [paraméterek]] [/NOPROF]

parancssorral indítható. A /NOPROF opció hatására hívás után nem indul el automatikusan a PROFILE.UNI parancsfile, egyébként igen. Ha parancsfile-nevet is megadtunk, akkor a PROFILE.UNI lefutása után a specifikált nevű parancsfile indul el automatikusan. Mind a PROFILE.UNI-t, mind a <parancsfile-név>.UNI-t az aktuális könyvtáron kívül a rendszer path-on is végigkeresi, ha nem találja, akkor "<parancsfile-név>.UNI : érvénytelen parancs" hibajelzést ad. A parancsfile-nak max. 5 hívási paramétert adhatunk meg, amelyeket a parancsfile-ból elérhetünk és feldolgozhatunk. A parancsfile-névnek nem kell tartalmaznia az .UNI kiterjesztést, mert ez alapértelmezés. Ha a parancsfile-ból nem a DOS-ba léptünk ki (1. QUIT parancs), az UNI> prompt jelenik meg a képernyőn, és parancsokat adhatunk ki a programnak.

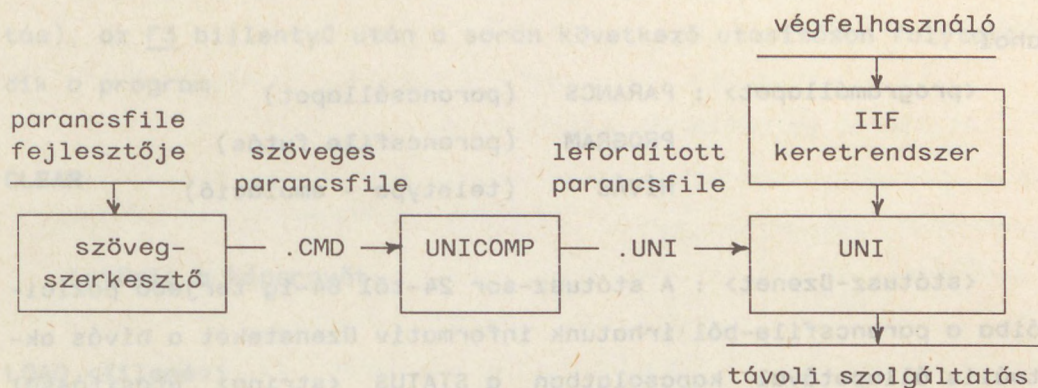
A parancsfile futása F9 vagy F10 billentyűkre félbeszakad: F9-cel az UNI parancsállapotba, F10-zel a DOS-hoz térhetünk vissza.

A program állapotait az alábbi ábra szemlélteti:



A hívási állapot a teletype-emulációt jelenti, amikor a felhasználó közvetlen kapcsolatban áll a hálózattal (amit begépel, az a hálózatra kerül ill. ami a hálózatról érkezik, az a képernyőre kerül ki). Hívási állapotban működik a karakter fordítás. Innen F3 billentyűvel léphetünk ki. Ekkor a parancsfile végrehajtása folytatódik, ill. UNI parancsállapotba kerülünk, attól függően, hogy honnan kértük a hívási állapotot. F10-re mindenképpen a DOS-hoz térhetünk vissza. Ezenkívül az escape-karakterrel különböző parancsokat adhatunk ki (az escape-karakter parancsfile-ből beállítható, alapértelmezésben a KERMIT-nél megszokott <CTRL>]): Az <escape-karakter> C hatása ugyanaz, mint az F3-é, az <escape-karakter> B egy break-et küld a vonalra, az <escape-karakter> F időlegesen a parancsállapotba lép ki, ahonnan a QUIT paranccsal visszatérhetünk a teletype-emulációhoz. (Ez utóbbi tulajdonság különösen parancsfile-ből indított teletype-emuláció esetén hasznos: Ha rájövünk például, hogy nincs nyitott naplófile, de nem akarunk kilépni a parancsfile-ből, akkor az <escape-karakter> F-re UNI parancsállapotba kerülünk, ahol a LOG OPEN <logfile-név> paranccsal megnyitjuk a naplófile-t, majd QUIT paranccsal visszatérünk a teletype-emulációhoz. FIGYELEM: Ha ilyen esetben a C paranccsal lépünk ki, az UNI program STACK OVERFLOW hibaüzenettel leáll!. Ha az UNI programot az IIF keretrendszerből indítottunk (és a végfelhasználó számára mindenképpen így a kényelmesebb), akkor a felhasználó az UNI parancsállapottal nem is találkozik. Az UNI program DOS-ból való indítása, a parancsállapot, és az UNICOMP program azok számára hasznos, akik különböző szolgáltatások elérésére szolgáló parancsfile-okat fejlesztenek és próbálnak ki.

A parancsfile-ok készítésének és használatának menetét az alábbi ábra mutatja:



2.1. A képernyő

Az UNI program a képernyő 25 sorából 24-et használ, a legalsó sor az ún. státusz-sor. A felső 24 sor megtelte után a képernyő rollozódik. Lehetőség van a lerollozódott információ visszalapozására is: a program minden képernyős kiírásból (akár teletype-emulációban, akár parancsállapotban vagy parancsfile-ből keletkezett) az utolsó 1024 karaktert eltárolja. Ha a program konzolinputra (parancsra vagy valamilyen karakterre) vár, a ↑, a ↓, a PgUp, a PgDn, a Home és az End karakterek hatására az eltárolt karaktereket előre-visszalapozhatjuk: ↑-al egy sorral feljebb, ↓-al egy sorral lejjebb, PgUp-pal 23 sorral feljebb, PgDn-nal 23 sorral lejjebb, Home-mal az 1024 karakteres buffer elejére, End-del a végére léphetünk. Továbbmenni (érvényes karaktert beírni) csak akkor lehet, ha a buffer végén állunk, egyébként a program csak lapozó karaktereket fogad el.

A státusz-sor az alábbi információkat tartalmazza:

<programállapot> <státusz-üzenet> <UNI verziószám>

UNI 2.0

ahol

<programállapot> : PARANCS (parancsállapot)
PROGRAM (parancsfile futás)
HÍVÁS (teletype - emuláció)

<státusz-üzenet> : A státusz-sor 24-től 64-ig terjedő pozícióba a parancsfile-ből írhatunk informatív üzeneteket a hívás aktuális állapotával kapcsolatban a STATUS <string> utasítással (pl. "A kapcsolat felvétele sikerült" vagy "A szolgáltatás nem üzemel")

<verziószám> : Az UNI verzió számát írja ki, pl. UNI 2.0. Parancsfile futása esetén ezen a helyen WAIT és PAUSE utasítás végrehajtása alatt egy másodpercenként növekvő számláló látható, ami azt jelzi, hogy a program "nem fagyott le", hanem valamilyen hálózati eseményre vár.

2.2. Az UNI program parancsállapotából kiadható parancsok

C (CONNECT) parancs

A teletype-emulációt hívja. Innen az F3 billentűvel lehet kilépni. Az <escape-karakter> B sorozat egy break-jelet küld a vonalra. Az escape-karakter programozható. (SET ESCAPE=&<hexa-kód> utasítás), alapértelmezés a KERMIT-nél megszokott CTRL-].

Ha a C parancs után a CARRIER OFF üzenetet kapjuk, akkor a vonal valamilyen ok miatt nem él (pl az RS232 kártya kivezetéséhez nincs kábel csatlakoztatva, stb).

Ha parancsfile-ból hívtuk a teletype-emulációt (TTY utasítás), az F3 billentyű után a soron következő utasításon folytatódik a program.

CLEAR

Letörli a képernyőt.

LOAD <filenév>

Betölti és elindítja a <filenév>.UNI parancsfile-t. Ha a file nem létezik, akkor a "<filenév>.UNI file megnyitási hiba" üzenetet kapjuk vissza. Hatása ugyanaz, mintha egyszerűen a <filenév> parancsot adnánk ki. A különbség csupán annyi, hogy ez utóbbival nem tudunk valamely parancssal egyező nevű parancsfile-t (pl. "QUIT.UNI") indítani (hiszen ezt a program kilépő parancsnak értelmezné), míg a LOAD QUIT parancsra az ilyen nevű file indul el.

LOG

Napló (log) file kezelő parancs. Az alábbi változatai vannak:

LOG OPEN <filenév> : Megnyitja a <filenév> naplófile-t.

LOG CLOSE : Lezárja a naplófile-t.

LOG SUSPEND : Felfüggeszti a képernyőn megjelenő információk naplózását

LOG RESUME : Ujraindítja a képernyőn megjelenő információk naplózását

UNI 2.0

SET

Az alábbi paraméterek állíthatók be:

SET	SPEED	300	: A vonali sebességet állítja be.
		600	
	BAUD	1200	
		2400	
		4800	
		9600	

Figyelem! A 300 baudos vonalak kezeléséhez a DEVIC.SYS driver helyett a DEV300.SYS drivert kell betölteni! (1. UNI installálása c. fejezet)

SET	PARITY	ODD EVEN NONE	: A paritást állítja be.
SET	ACCENT	ON OFF	: Az ékezetes karakterfordítást kapcsolja ki-be.
SET	TRACE	ON OFF	: A parancsfile-végrehajtás nyomkövetését kapcsolja ki-be
SET	LOCAL	ON OFF	: Hívási állapotban a begépelte karakterek helyi visszairását kapcsolja ki-be. Alapértelmezésben ki van kapcsolva, bekapcsolt állapot esetén a kettős visszairás elkerülése érdekében gondoskodni kell arról, hogy az ellenoldal ne írja vissza ("ekhözza") a karaktereket

QUIT

Végleges kilépés a DOS-ba.

3. Az UNI parancsfile-nyelv leírása

Az UNI parancsfile-ok minden esetben .CMD kiterjesztésű szövegfile-ok, amelyeket tetszőleges szövegszerkesztő programmal hozhatunk létre. A szöveges parancsfile-t az UNICOMP programmal használat előtt le kell fordítani .UNI kiterjesztésű, az UNI által végrehajtható formátumú parancsfile-lá. A parancsnyelv egy saját fejlesztésű protokoll-leíró nyelv, amely alkalmas adaptív hívási-bejelentkezési protokollok leírására. A parancsfile-okban azokat a hívási protokollokat kell leírni, amelyeket egyébként is lejátszanánk "kézi" hívás esetén. Ezért a parancsfile-ok megfogalmazásához alapfokú programozási ismereten kívül hálózati ismeretek is szükségesek (hívási címek, PAD paraméterek stb.). A parancsfile-ok belövését segíti a nyomkövetési lehetőség (SET TRACE ON), amikor minden végrehajtott UNI utasítás eredménye megjelenik a képernyőn és a parancsfile végrehajtása utasításonként SPACE karakterrel folytatható. Így ellenőrizhető, hogy a program a kívánt logikai ágon fut-e, megérkezett-e a kívánt válasz az ellenoldaltól ill. mi jött helyette stb.

A nyelv főbb jellemzői a következők:

- Struktúrája hasonló a hagyományos programozási nyelvekéhez: megtalálhatók benne a feltételes utasítások, ciklusutasítások, stringváltozók, szubrutinhívások, bemeneti-kimeneti műveletek stb. Így alapfokú programozási ismeretek (pl. a BASIC nyelv ismerete) is elegendőek a parancsfile-ok megfogalmazásához.

UNI 2.0

- Lehetővé teszi az eseményvezéreltség elvének alkalmazását, váratlan hálózati események parancsfile-ból kezelhetőek (pl. váratlan vonalelbomlás esetén a hívást újra megkísérelheti stb.)

- Adaptív hívási protokollok is leírhatók: ha egy szolgáltatás eléréséhez több lehetséges hívási út is van, és az első nem áll rendelkezésre, a másikon próbálkozhat, illetve a felhasználót megfelelően tájékoztatja a sikertelen hívás okáról.

A parancsfile-ban egy szövegsor egy utasítást tartalmaz és az alapszavakban, változónevekben, konstansnevekben a kis- és nagybetűk között nincs különbség. A *-gal kezdődő sorok megjegyzéssorokat jelentenek. A sor elején tetszőleges számú SPACE karakter állhat, így a parancsfile külső képe a programstruktúrának megfelelő módon alakítható ki.

A parancsfile-nyelv a program későbbi fejlesztése során - a kompatibilitás megtartásával - bővílni fog.

3.1. Stringkonstansok

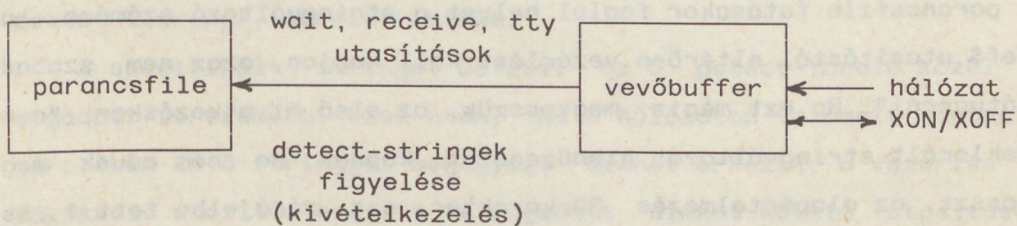
Stringkonstans két " jel közé zárt karaktersorozat. Itt escape-karakterként szerepel a & (hexa-karakter jel) és a ^ (decimális karakter jel). Ha a stringben ilyeneket is akarunk szerepeltetni, kettőzni kell őket. A &<hexa-byte> egy hexadecimálisan, a ^<decimális-byte> egy decimálisan adott kódú ASCII karaktert jelent. Tehát a "&0d^10" stringkonstans egy <CRLF> sorozatot jelent, a "&05&&QW" pedig egy "<CTRL-E>&QW" sorozatot. Stringek összefűzhetőek a ! jellel: pl. a


```
"PASSWORD:" | "&0d"
```

stringkonstans egy ENTER-rel lezárt karaktersorozatot jelent. Egy stringkonstans maximális hossza 255 karakter lehet.

3.2. A vevőbuffer

A vonalon érkező karaktereket az UNI program egy bufferben, az ún. vevőbufferben tárolja. A vevőbuffer túlcsordulását elkerülendő, a vonali vezérlőprogram automatikusan elvégzi az XON/XOFF protokollal kapcsolatos teendőket. A parancsfile-ból a vevőbuffer tartalmához férhetünk hozzá. A vevőbuffer kezelése az alábbi ábrán látható:



3.3. Az UNI parancsnyelv utasításai

Értékadó utasítás

<stringváltozó>=<string>, ahol a stringváltozónak előzőleg deklaráltnak kell lennie. A <string> konkatenálással összeállított stringelemekből is állhat.

Példa:

```
chr password(8)
password="Hello" | enter
```

break

Egy break-jelet küld a vonalra.

call <címke>

Szubrutinhívás a <címke> utáni utasításra. A szubrutin a **return** utasítással ér véget. A szubrutinok max. 15-ös mélységben ágyazhatók egymásba, utána "stack túlcsordulás" hibaüzenetet kapunk. A címke mindig ":" jellel kezdődik, és tetszőleges alfanumerikus karakterekből álló sorozat lehet.

chr <változónév>[(<hossz>)]

Stringváltozót deklarálni. Ezek után a stingváltozó értékadó, paraméterátvevő vagy bemeneti utasításon keresztül értéket kaphat, és stringek leírásánál szerepelhet. Minden stringváltozót deklarálni kell az első előfordulása előtt. Mivel a chr utasítás a parancsfile futásakor foglal helyet a stringváltozó számára, a def\$ utasítástól eltérően vezérlést kell kapjon, azaz nem szabad "átugrani". Ha ezt mégis megtesszük, az első hivatkozáskor "nem deklarált stringváltozó" hibaüzenetet kapunk. Ha nem adunk meg hosszt, az alapértelmezés 80 karakter; ezt zárójelbe tett 1 és 127 közötti számmal felülbírálnhatjuk. Értékadáskor a hosszt nem léphetjük túl, ellenkező esetben a "Túl hosszú string" hibaüzenetet kapjuk és a parancsfile végrehajtása megszakad. A deklarált stringváltozók együttes mérete nem haladhatja meg az 1024 byte-ot.

Változónév alapszó nem lehet, egyébként alfanumerikus jeleket (az angol ABC betűit és számjegyeket) és "_" karaktert tartalmazhat, minden esetben betűvel kell kezdődnie. Tetszőleges hosszú lehet, de a fordító csak az első 8 karakterben tesz különbséget. Stringváltozónévben nincs különbség a kis-és nagybetűk között, tehát például a ValaszString és a VALASZSTrinG azonos változókat jelentenek.

def\$ <konstansnév>=<stringkonstans>

Az adott konstansnévhez rendeli a stringkonstanst. A névadásra ugyanazok a megszorítások vonatkoznak, mint a stringváltozó-nevekre, ezenkívül stringkonstansnak és stringváltozónak nem lehet azonos neve. A definiált stringkonstansra ezután a névével is hivatkozhatunk.

Példa:

def\$ enter="^13"	(konstansdefiníciók)
def\$ default_pw="TITOKZATOS!"	
chr password(25)	(változódeklaráció)
password = default_pw enter	(értékkadás)

A def\$ utasítás deklaratív jellegű, azaz nem feltétlenül kell vezérlést kapnia, "átugorható".

detect <string>,<tároló>,<címke>

A specifikált stringet beteszi az 5 detect-tároló közül a megadott sorszámúba. Ezek után, ha a hálózaton valamely detect-tárolóban lévő stringgel megegyező üzenet érkezik, a vezérlés a következő utasítás helyett a megadott címkét követő utasításon folytatódik. Így valósítható meg a kivételkezelés, pl. a vonal elbomlásának figyelése. Az X25-ös PAD pl. vonal elbomlásakor egy CLR DTE (0,0) üzenetet küld. Ez a

```
detect "CLR DTE (0,0)",1,lebomlott
```

utasítással figyelhető, és az üzenet megérkezésekor a program a ":lebomlott" címkéjű soron folytatja a vezérlést, ahol a vonalelbomlás kezeléséről intézkedhetünk.

A detect-stringeket az UNI csak receive, wait és tty utasítások végrehajtása alatt figyeli. A specifikált szöveg megérkezése után a vevőbuffer a specifikált szöveget már nem fogja tartalmazni. Ha ugyanarra a tárolóra több detect utasítást aduk ki, az utolsó lesz érvényes.

end

A parancsfile-ből UNI parancsállapotba lép ki. Ha nem szerepel a programban, az utolsó utasítás után automatikusan megtörténik.

find <string₁> <string₂>

A <string₁>-ben megkeresi a <string₂> első előfordulását, ha van, és az eredménytől függően beállítja a found rendszerfeltételt, amelyre feltételes utasításokból hivatkozhatunk.

Példa:

```
chr buffer(127)
receive buffer
find buffer,"TTY01"
if found
    say "Van benne!"
else
    say "Nincs benne"
endif
```

goto <címke>

Ugrás a <címke> utáni utasításra.

Példa:

```
goto exit
.
.
:exit
quit
```

```
if <feltétel>
```

```
    utasítások
```

```
[else
```

```
    utasítások          ( Ez a rész opcionális! )
```

```
    ]
```

```
endif                  ( Kötelező )
```

Ahol <feltétel>:	waitfor		Ezeket a wait utasítás állítja
	timeout		be.
	found		Ezt a find utasítás állítja be

vagy <feltétel>:	<string ₁ >		=		<string ₂ >
			<>		

Ez utóbbi a két stringet összehasonlítja és egyezésük esetén a feltétel igaz értékű lesz.

Az ELSE ág szerepeltetése nem kötelező, de az ENDIF utasításé igen!

Példa:

```
receive valasz
if valasz<>"ENTER PASSWORD"
    say "Nincs kapcsolat"
    quit
else
    transmit password_string ! enter
endif
```

include <string>

A stringben megadott nevű parancsfile-t (a névben a ".CMD"-nek nem kell szerepelnie) "belefordítja" a parancsfile-ba. Hatása ugyanaz, mintha a parancsfile-t szövegszerkesztővel belemásolnánk az eredeti parancsfile megfelelő helyére. Az **include** utasítások max. 5 mélységben ágyazhatók egymásba.

input <stringváltozó>

A billentyűzetről bevett karaktersorozatot a specifikált stringváltozóba helyezi el. Javításra a BACKSPACE karakter használható, a lezáró ENTER nem lesz benne a stringváltozóban.

killdetect <tároló-sorszám>

Az adott sorszámú tárolóban lévő **detect** string figyelését letiltja.

logopen <string>

Megnyitja a <string> nevű naplófile-t. Nyitott naplófile esetén ami a képernyőre kerül, egyben a naplófile-ba is belekerül. Ez igen hasznos, ha pl. adatbázislekérdezésnél a lekérdezés eredményét utólagos felhasználás, kinyomtatás céljából el akarjuk tárolni.

logclose

Lezárja a korábban megnyitott naplófile-t

minput <stringváltozó>

Az input utasításhoz hasonló, de nem írja vissza a bevett karaktereket, ezért jelszó bekéréséhez használható.

Példa:

```
chr password(10)
wait "ENTER PASSWORD"
if waitfor
  say "Jelszó:"
  minput password
  transmit password!enter
endif
```

on_croff <címke>

A detect utasítás után a második módja a kivételkezelésnek. Ha a hálózat ún. carrier-jele megszűnik, az itt specifikált utasításon folytatódik a vezérlés.

parse <stringváltozó>

A parancsfile paramétereit veheti át.

Példa:

```
chr adatb(8)
parse adatb
transmit "ISIH " !uid! " !pwd! " !adatb!enter
```

Ha ezt a parancsfile-t a <parancsfile-név> KUFET parancssorral indítjuk UNI parancsállapotból vagy az

UNI <parancsfilenév> KUFET

parancssorral DOS-ból, a fenti utasításokra az "adatb" változóba bekerül a paraméterként adott "KUFET" szöveg. A következő parse utasítás az esetleges következő paramétert veszi át; ha nincs ilyen, akkor az itt specifikált stringváltozó értéke üres lesz.

pause <szám>

A paraméterként adott számú másodpercig vár. Pl. a pause 5 utasítás 5 másodpercig vár, és a képernyő státusz sorában a verziószám helyett a "SZÜNET <számláló>" szöveg jelenik meg, és a számláló értéke másodpercenként változik 1-től 5-ig.

receive <stringváltozó>

A vonali vevőbuffer tartalmát maximum az adott stringváltozó-
nak megfelelő hosszban beolvassa a stringváltozóba. Ha a vonali
vevőbuffer üres, a stringváltozó értéke üres lesz.

rectrans <filenév>

A vételi karakterfordítási táblát aktivizálja. Ez egy
szöveges file, ahol a következő formájú sorok jelentik a bejegy-
zéseket:

```
"<stringkonstans1>"="<stringkonstans2>"
```

Ekkor a vonalon érkező minden "stringkonstans₁" karaktersoro-
zatot "stringkonstans₂"-vé fordít le. Pl. a fordítási file
"@7A"="^143" sora hatására a vonalon érkező minden "@7A" karak-
tersorozat teletype-emulációnál "Á"-vá fordul le (az "Á" decimá-
lis kódja 143). Maximum 32 fordítási sor adható meg. Ha a fordí-
tási file nem található meg az aktuális könyvtárban, a rendszer
path-on is végigkeresi. Ha itt sem találja, akkor a "<filenév>
megnyitási hiba" hibaüzenetet kapjuk.

return

Visszatérés a szubrutinból. Ha nem szubrutinhívó utasítást
követően kerül return utasításra a vezérlés, a "stack alulcsordu-
lás" hibaüzenetet kapjuk.

rflush

A vonali vevőbuffert törli ki.

quit

A DOS operációs rendszerhez (illetve az IIF keretrendszerhez)
tér vissza.

say <string>

Az adott stringet kiírja a képernyőre.

sendtrans <filenév>

Az adási karakterfordítási táblát tölti be. A file formátuma hasonló a vételi fordítási file-hoz.

set

Set paranccsal az alábbi paraméterek állíthatók be:

set baud	300	: A vonali sebességet állítja be.
	600	
	1200	
	2400	
	4800	
	9600	

Figyelem! A 300 baudos vonalak kezeléséhez a DEVIC.SYS driver helyett a DEV300.SYS drivert kell betölteni! (1. "UNI installálása" c. fejezet)

set parity	odd	: A vonali paritást állítja be.
	even	
	none	

set accent	on	: Az ékezetes karakterfordítást kapcsolja ki-be.
	off	

set trace	on	: A parancsfile végrehajtás nyomkövetését kapcsolja ki-be
	off	

set local	on	: Hívási állapotban a begépett karakterek helyi visszairását kapcsolja ki-be.
	off	

set escape=&<hexa-kód> : az escape karakter beállítására szolgál. A set escape=&1b utasítás az escape karaktert a KERMIT programban megszokott <CTRL> "]"-re állítja be.

status <string>

A státusz-sorba írja a string első 45 karakterét. Itt jelezhető, hogy hol tart a host felhívása és a bejelentkezés ill. milyen váratlan események történtek. (pl. status "A kapcsolat lebomblott - újra megpróbálok")

time <szám>

A belső órát (timeout-érték) a megadott számú másodpercre állítja be. A belső órát a a wait utasítás vizsgálja. A pause és a time utasítások ugyanazt a belső órát használják, így a time után kiadott pause utasítás elállítja az órát!

Példa:

```
time 20          (Itt a timeout 20!)
.
.
pause 1
wait "CONNECTED" (Itt a timeout már 1!)
```

transmit <string>

A stringet kiküldi a vonalra.

Példa:

```
def$ enter "^13"
transmit "SELECT,2801015020"
transmit enter
```

tty

A teletype-emulációt hívja, ahonnan az F3, F9 vagy F10 billentyűkkel lehet kilépni:

F3 : Visszatérés a parancsfile folytatásához

F9 : Visszatérés UNI parancsállapotba

F10: Visszatérés a DOS-hoz (vagy a keretrendszerhez)

Teletype-emuláció alatt a státusz-sorban a "Kilépés : F3" szöveg látható.

wait <string>

Az előzőleg beállított timeout értékig várakozik a megadott stringre. Ha bejött, a waitfor feltétel igaz lesz, ha nem, akkor hamis. Várakozás alatt a státusz-sor jobb oldali részében, a verziószám helyén a VÉTEL <számláló> szöveg látható, ahol a számláló értéke 1-ről indul és másodpercenként növekszik mindaddig, amíg a várt string meg nem érkezett, vagy a timeout le nem járt. A timeout feltétel minden esetben a waitfor logikai negáltja. A várt szöveg megérkezése után a vevőbuffer a maradékot fogja tartalmazni, magát a szöveget már nem.

while <feltétel>

. (utasítások)

endwhile

Amíg a specifikált feltétel teljesül, a közbezárt utasításokat ciklikusan végrehajtja. A <feltétel> ugyanaz lehet, mint az if utasításban.

3.4 Példaprogram

Ezek után nézzünk egy példaprogramot! Ez a parancsfile a VEIKI számítógépén működő "IIF központi szolgáltatás" rendszerbe való bejelentkezést hajtja végre.

```

goto start
* Itt vannak leírva a felhasználóspecifikus információk!
include "user"
:start
chr password(10)
detect "ALREADY LOGGED ON",1,busy
status "Tűrelmet kérek"
say "*****"
say "Belépés a központi szolgáltatás adatbázisba"
say ""
say "      (VEIKI IBM - VM/CMS)"
say ""
say "USERID: " !veiki_userid
say "*****"
say ""
*
* X25 kapcsolat felvétele
*
call connect
say "VEIKI CMS jelszó:"
minput password
* Újraindítási pont
:pad
transmit "select,2801015020"
transmit enter
detect "CLR DTE",2,bomlott
time 3
wait "COM"
if waitfor
    time 10
    transmit enter
    wait "LOGOFF"
    if waitfor
        status "Userid:" !veiki_userid
        transmit "l " !veiki_userid !enter

```

```

wait "PASSWORD"
if waitfor
  transmit passwordlenter
  detect "INCORRECT PASSWORD",3,invalid
  time 40
  status "Kapcsolat él! - bejelentkezés folyik"
  wait "ker"
  if waitfor
    transmit "I"|lenter
    wait "23"
    if waitfor
      transmit enter
      wait "I/N"
      if waitfor
        transmit "i"|lenter
        status "Napló: VEIKI.LOG      Kilépés: F3"
        logopen "veiki.log"
        tty
        killdetect 2
        logclose
        transmit "logoff"|lenter
        status "LOGOFF - KILÉPÉS FOLYAMATBAN VAN"
        time 20
        wait "CLR"
        if waitfor
          status "LOGOFF - KILÉPÉS A VEIKI HOSTBÓL"
        endif
      endif
    endif
  endif
endif
else
  status "Nincs kapcsolat a VEIKI hosttal"
endif
else
  status "A postai vonal nem áll rendelkezésre"
endif
break
transmit break_str
quit

:busy
status "Ez a virtuális gép foglalt!"
quit

:bomlott
killdetect 2
status "A kapcsolat lebomlott - türelmet kérek!"

```

UNI 2.0

```
goto pad
quit

:invalid
status "Érvénytelen jelszó!"
transmit ctrl_p
transmit "clr&0d"
quit

:connect
if network="LAN"
    break
    transmit break_str
    wait "/"
    if waitfor
        transmit "gt"!enter!enter
        time 3
        wait "*"
        if waitfor
            return
        else
            status "A WAN hálózat nem érhető el!"
            quit
        endif
    else
        status "A LAN hálózat nem érhető el"
    endif
else
    transmit enter
    wait "*"
    if waitfor
        return
    else
        status "A hálózat nem érhető el!"
        quit
    endif
endif
endif
```

4. Az UNI installálása

Az "IIF keretrendszer" és az "UNI" programrendszerek azonos lemezen kerülnek forgalomba. Installálásukhoz egy megfelelő directory készítése után egyszerűen a COPY A:*. * parancsot kell kiadni. Az UNI program működéséhez szükséges még a DEVIC.SYS driver installálása is. (A 300 baudos hálózathoz kapcsolódó felhasználónak a DEV300.SYS drivert kell installálni! Ekkor a

SET BAUD 300 parancs lesz az egyedül érvényes vonali sebességbe-állító parancs, ezt vagy a PROFILE-ban, vagy mindegyik parancsfile-ban ki kell adni!) A driver installálása azt jelenti, hogy a CONFIG.SYS file-nak tartalmaznia kell egy

```
DEVICE = [ pathnév ] DEVIC.SYS
```

vagy egy

```
DEVICE = [ pathnév ] DEV300.SYS
```

sort.

Installálás után mind az IIF rendszer MAINPAGE.IIF file-ja, mind az UNI parancsfile készlete tetszőlegesen, az egyéni igényeknek, valamint a folyamatosan bővülő szolgáltatás-készletnek megfelelően bővíthető.

A disztribúciós lemezen lévő keretrendszerből hálózati szolgáltatásként a VEIKI "IIF központi szolgáltatás" rendszere és az MTA SZTAKI IBM 148-as számítógépén működő ISIS adatbázisok érhetőek el. Az USER.CMD file tartalmazza azokat az alapadatokat, amelyek felhasználónként és munkaállomásonként különböznek. Ilyenek a VEIKI IBM CMS felhasználói azonosító("veiki_userid"), az ISIS felhasználói azonosító ("isis_userid"), a vonali sebesség ("baud"), a C.I.C.S terminálazonosító ("isistwx"), a hálózat jellege ("network=LAN" vagy "network=PAD"), LAN hálózat esetén az a karakter, amelyre a LANP BOX egy BREAK jel után bontja a kapcsolatot ("break_str") stb. Első közelítésben ezeket az információkat kell a valós helyzetnek megfelelően átírni, majd a megfelelő parancsfile-okat újra lefordítani (UNICOMP VEIKI és UNICOMP IBM148). Ezek után a keretrendszer és az UNI használható lesz. Amennyiben újabb szolgáltatásokat akarunk elérni, az elérés módját UNI parancsfile-ban leírva, a parancsfile-t lefordítva és az IIF keretrendszer menüleíró file-ját megfelelően bővítve a szol-

UNI 2.0

gáltatás elérhető lesz. Lokális jellegű szolgáltatások (szöveg-szerkesztők, lokális adatbáziskezelők stb.) esetén természetesen csak a menüleíró file-t kell bővíteni az IIF KERETRENDSZER dokumentációban leírtak szerint.

4.1. A disztribúciós lemez tartalma

A mindenki által megkapott IIF feliratú lemez (360 K, DS/DD) az IIF keretrendszert és az UNI programot tartalmazza. Az UNI programhoz mellékelte minta parancsfile-ok a parancsnyelv könnyebb megtanulását segítik.

Az IIF keretrendszerhez az alábbi file-ok tartoznak:

IIF.COM : Maga az IIF keretrendszer program
IIF.IIF : A bejelentkező ábra
PROFILE.IIF : Az alapparamétereket beállító parancsfile
MAINPAGE.IIF : A menülapleíró file

Az UNI rendszerhez az alábbi file-ok tartoznak:

UNI.EXE : Az UNI program
UNICOMP.EXE : Az UNICOMP program
DEVIC.SYS : Az UNI program működéséhez szükséges vonali DOS device driver.
DEV300.SYS : A 300 baudos hálózathoz kapcsolódó felhasználók számára ez a device driver szükséges.
GIZMO.IN : A bemeneti GIZMO -> ékezetes fordítást leíró file
GIZMO.OUT : A kimeneti ékezetes -> GIZMO fordítást leíró file
PROFILE.CMD és

PROFILE.UNI : Az alapparaméterek beállítása
USER.CMD : Alapinformációkat tartalmazó file.
Ezt installáláskor megfelelően módosítani kell.
PROC.CMD : A CONNECT szubrutint tartalmazza
VEIKI.CMD és
VEIKI.UNI : A VEIKI hostba (központi szolgáltatás) való
belépést írja le
IBM148.CMD és
IBM148.UNI : Az MTA SZTAKI 148-as számítógépén üzemelő adat-
bázisokba való bejelentkezést írja le. Hívása:
IBM148 <adatbázisnév>.

UNI 2.0

=====

 MUTATÓ

 =====

B
 break 6, 8, 14, 27

C
 CARRIER 8, 19

D
 detect 15, 18, 19
 DEVIC.SYS 10, 21, 26, 28

E
 end 16
 escape-karakter 6, 8, 22
 eseményvezéreltség 12

É
 ékezetes 1, 2, 3, 10, 21, 28

F
 find 16
 found 16, 17

G
 GIZMO 3, 28
 goto 16

I
 if 17
 IIF 1, 6, 21, 26, 27, 28
 include 18
 input 18

K
 karakterfordítás 1, 2, 3,
 10, 20, 21
 képernyő 3, 6, 7, 21
 killdetect 18
 kivételkezelés 19

M
 minput 18

N
 naplófile 3, 6, 9, 18

P
 parancsállapot 5, 6, 7, 16
 parse 19
 PAUSE 8, 20, 22
 PROFILE 5, 27, 28

Q
 QUIT 5, 6, 10, 21

R
 receive 15, 20
 rectrans 3, 20
 return 14, 20
 rflush 20

S
 say 21
 sendtrans 3, 21
 SET 4, 8, 10, 21, 26
 státusz-sor 7, 20, 22, 23
 string 12, 13, 21, 22, 23
 stringkonstans 12, 15
 stringváltozó 13, 14, 15,
 18, 19, 20

T
 terminál-emuláció 2, 6, 8,
 9, 20, 23
 time 22
 timeout 17, 22, 23
 transmit 22
 TTY 9, 15, 23

U
 UNI 1, 2, 3, 4, 11, 13,
 26, 27, 28
 UNICOMP 2, 4, 11, 27

V
 vevőbuffer 13, 15, 20, 23
 visszalapozás 3, 7

W
 WAIT 8, 15, 22, 23
 waitfor 17, 23
 while 23

UOI DONALI IRASHIBA

53

ACCENT OFF bell.

VEIKI 02161280101 6558

02161~~0~~280100 6832

PETRA.out ?

6820