

1. UVOD

UPORABNIK TERMINALA, PRIKLJUČENEGA NA RAČUNALNIK, OBIČAJNO NE SME (IN TUDI NE ŽELI) IMETI DOSTOPA DO POLJUBNIH RAČUNALNIŠKIH PROGRAMOV IN UKAZOV. RAVNO TAKO ŽELI NA ENOSTAVEN NAČIN PROŽITI PROGRAME, KI JIH UPORABLJA PRI SVOJEM DELU, BREZ TIPKANJA ZAMUDNIH UKAZOV, KOT JE NA PRIMER RUN DR1:Š140,276ČA01CRE. RAČUNALNIK MORA BITI ZAŠČITEN PRED MOŽNOSTJO, DA NEK UPORABNIK Z NAPAČNO PODANIM UKAZOM NA PRIMER ZBRIŠE NEKAJ LET USTVARJANO PODATKOVNO BAZO Z MAGNETNEGA DISKA, ITD. PROGRAMSKI PAKET PRG NAM REŠUJE TE PROBLEME. OMOGOČA HKRATI POPOLNO ZAŠČITO PRED NEZAŽELJENIMI POSEGI V RAČUNALNIŠKI SISTEM KOT TUDI ENOSTAVEN NAČIN PROŽENJA PROGRAMOV OZIROMA OBDELAV.

2. UPORABA

DA BI S PROGRAMSKIM PAKETOM PRG LAHKO ZAGANJALI RAZNE PROGRAME OZIROMA PROCEDURE, MU MORAMO SEVEDA DEFINIRATI NABOR (MENU) ŽELJENIH OBDELAV. TEGA PREPROSTO SESTAVIMO Z EDITORJEM, TAKO, DA V DATOTEKO, KATERE IME NAJ IMA VEDNO POLJUBNIH 6 ZNAKOV, EKSTENZIJA PA NAJ BO ".PRG", RECIMO "SEZNAM.PRG" (EKSTENZIJA JE LAHKO TUDI DRUGAČNA, VENDAR JO MORAMO TEDAJ PRI UPORABI PRG PAKETA PODAJATI, KER PAKET SICER AVTOMATSKO PRIVZAME ".PRG"). VSAKA VRSTICA TE DATUTEKE V SVOJEM PRVEM DELU NOSI TEKST, KI SE IZPIŠE V MENUJU, PRIKAZANEM NA ZASLONU IN PRIPADAJUČEGA POLJUBNEGA UKAZA, KI GA RAČUNALNIK RAZUME. MED TEKSTOM IN PRIPADAJOČIM UKAZOM MORA OBVEZNO STATI ZNAK "#". V EN MENU LAHKO VNESEMO NAJVEČ 40 UKAZOV. NA ZASLONU SE PRIKAŽEJO PRVI V LEVEM DELU PRVE VRSTICE MENUJA, DRUGI V DESNEM DELU PRVE VRSTICE MENUJA, TRETJI V LEVEM DELU DRUGE VRSTICE, ITD. DA BI UKAZE LAHKO STARTALI, MORA IMETI OPISNI TEKST NA PRVIH DVEH MESTIH OBVEZNO DVOMESTNO ŠTEVILO OD 01 DO 40, KI OZNAČUJE UKAZ. SEVEDA PREJ OMENJENA OMENJENIH MAKSIMALNIH 40 UKAZOV NE PREDSTAVLJA OMEJITVE. KER OB INSTALACIJI PROGRAMSKEGA PAKETA STARTANJE MENUJEV POSTANE RAČUNALNIŠKI UKAZ, SE LAHKO PRVI NABOR SESTOJI IZ 40 UKAZOV ZA ŠTARTANJE 40 RAZLIČNIH NOVIH NABOROV, KAR NAM DA NA DRUGEM NIVOJU ŽE 1600 MOŽNOSTI. LAHKO PA GREMO ŠE NA TRETJI NIVO ...

MENUJSKO VODENJE OBDELAV STARTAMO PREPROSTO Z UKAZOM PRG XXXXXX, KJER XXXXXX POMENI DATOTEKO, KJER JE ZAPISAN USTREZEN NABOR. KO NA ZASLONU DOBIMO DEFINIRANEMU NABORU PRIPADAJOČO ZASLONSKO SLIKO (ALI IZPIS NA HARD COPY TERMINALU), LAKO Z VNOSOM DVOŠTEVILČNE OZNAKE UKAZA LE-TEGA SPROŽIMO, NA PRIMER, POŽENEMO NEK PROGRAM. KO PROGRAM DELA, SE MENU POBRIŠE Z ZASLONA, Z ZASLONOM PA DELA POGNANI PROGRAM. KO PROGRAM KONČA, SE NA ZASLONU ZOPET PRIKAŽE MENU, IZ KATEREGA JE BIL PROGRAM SPROŽEN. IZSTOP IZ NEKEGA MENUJA (TIPKA <ESC>) NAS POSTAVI V MENU NA NIŽJI NIVO, IZ KATEREGA JE BIL NAŠ MENU STARTAN, ČE PA SMO ŽE NA NAJNIŽJEM NIVOJU, IZSTOP IZ MENUJSKEGA VODENJA OBDELAV.

1
2
3 NAREDIMO SI PRIMER NABORA PROGRAMOV IN SI OGLEDAMO PRIPADAJOČO
4 ZASLONSKO SLIKO, KI JO PAKET PRG NAREDI NA ZASLONU. VZEMIMO, DA
5 ŽELIMO IZ PRVEGA MENUJA POGNATI 2 NASLEDNJA MENUJA, ALI POGNATI
6 PREPIS DIRECTORYJA Š130,6Č V REZERVO NA TRAK S PROGRAMOM BRU, ALI PA
7 NAREDITI PREPIS IZ REZERVE NA TRAKU V DIRECTORY Š130,6Č, NA DRUGEM
8 NIVOJU PA IMAMO V ENEM SEZNAMU PROGRAME KADROVSKE EVIDENCE, V DRUGEM
9 PA PROGRAME OBDELAVE OSEBNIH DOHODKOV. TOREJ KREIRAMO TRI NABORE
10 PROGRAMOV (3 .PRG) DATOTEKE, KI SE NAHAJAJO NA DR1:Š130,1Č

11
12 - NIVO10.PRG - PRVI SEZNAM

13 01 KADROVSKA EVIDENCA#PRG DR1:Š130,1ČNIVO21
14 02 OSEBNI DOHODKI#PRG DR1:Š130,1ČNIVO22
15 03 PREPIS Š130,6Č V REZERVO#BRU/REW/VER/MOU DR1:Š130,6Č*. * MTO:
16 04 PREPIS Š130,6Č IZ REZERVE#BRU/REW/VER MTO:Š130,6Č DR1:

17
18 - NIVO21.PRG - DRUGI SEZNAM (PERSONALNI PODATKI)

19 01 VNOS PODATKOV O DELAVCU#RUN DR1:Š130,2ČAMPERS
20 02 VNOS PRETEKLIH ZAPOSILITEV#RUN DR1:Š130,2ČAMZAPO
21 03 PREGLED DELAVCEV PO ABECEDI#RUN DR1:Š130,2ČASPEAB
22 03 PREGLED DELAVCEV PO ŠIFRI#RUN DR1:Š130,2ČASPESI

23
24 - NIVO22.PRG - TRETJI SEZNAM (PODATKI ZA ODO

25 01 VNOS OBCIN IN KRAJ.SKUPN.#RUN DR1:Š130,2ČAMOBCEI
26 02 VNOS PRISPEVKOV#RUN DR1:Š130,2ČAMPRIŠ
27 03 VNOS SAMOPRISPEVKOV#RUN DR1:Š130,2ČAMSAMO
28 04 VNOS POSOJIL#RUN DR1:Š130,2ČAMPOSO
29 05 VNOS EVIDENCE DELA#RUN DR1:Š130,2ČAPEVID

30
31 07 PREGLED POSOJIL#RUN DR1:Š130,2ČASPOSO
32 08 PREGLED KUMULATIV#RUN DR1:Š130,2ČASKUMU

33
34 OPOMBA: PRAZNA VRSTICA V .PRG DATOTEKI POMENI PRAZNO PRIPADAJOČE
35 MESTO NA ZASLONU.

36
37 Z UKAZOM PRG DR1:Š130,1ČNIVO1) DOBIMO NASLEDNJI ZASLON:

38
39 -----
40 ! S E Z N A M P R O G R A M O V (1) !

41 !
42 ! 01 KADROVSKA EVIDENCA 02 OSEBNI DOHODKI !
43 ! 03 PREPIS Š130,6Č V REZERVO 04 PREPIS Š130,6Č IZ REZERVE !

44 !
45 !
46 !
47 !
48 !
49 !
50 ! PODAJ ŠTEVILKO ŽELENEGA PROGRAMA (N+ŠTEV.ZA NAVODILA): __ !
51 !
52 -----

53
54 Z VNOSOM OZNAKE 03 SE NAM ZDAJ IZVEDE PREPIS V REZERVO, NAKAR SE NA
55 ZASLON VRNE NARISANI MENU (OZNAKA V OKLEPAJU POMENI NIVO, NA KATEREM
56 SE NAHAJAMO. S PRITISKOM NA <ESC> BI IZSTOPILI IZ MENUJSKEGA VODENJA
57 OBDELAV. Z VNOSOM 02 SE NAM PRIKAŽE NASLEDNJI ZASLON:
58
59
60
61
62
63
64
65
66
67
68
69
70

! PROGRAM AMOBICI UPORABLJAMO ZA PREGLED IN AŽURIRANJE OBČIN, !
! KRAJEVNIH SKUPNOSTI, REGIONALNIH ZAVODOV IN PREJEMNIKOV !
! ČLANARIN. POSAMEZNE SKUPINE MED SABO LOČIMO PO OBLIKI ŠIFRE. !
! TA JE ŠTIRIMESTNA IN IMA NASLEDNJO OBLIKO (X POMENI NUMERIČEN !
! ZNAK): !
! - XXXX - OBČINA !
! - ZXXX - REGIONALNI ZAVOD !
! - SIXX - PREJEMNIK SINDIKALNE ČLANARINE !
! - ZKXX - PREJEMNIK ZK ČLANARINE !
! !
! !
! I-IZVAJANJE, <RETURN>-NAPREJ, <ESC>-SEZNAM: _ !

ZDAJ LAHKO IZBIRAMO ALI TAKOJŠNJE IZVAJANJE (VNESEMO ČRKO I), ALI
LISTANJE NASLEDNJE STRANI NAVODIL (PRITISNEMO TIPKO <RETURN>), ALI
PA POVRATEK V MENU (PRITISNEMO TIPKO <ESC>).

3. ZAŠČITA PRED NEZAŽELJENIMI POSEGI V RAČUNALNIK

ČE UPORABNIK PAKET PRG STARTA Z UKAZOM PRG XXXXXX, KO JE ČE
PRIJAVLJEN NA RAČUNALNIŠKI SISTEM, NE MOREMO GOVORITI O KAKRŠNI KOLI
ZAŠČITI, SAJ, PREDNO POKLIČE MENUJE, ALI, KO IZSTOPI IZ NJIH, LAHKO
IZVAJA POLJUBNE RAČUNALNIŠKE UKAZE. ZATO IMA PROGRAMSKI PAKET PRG
MOŽNOST, DA SE AVTOMATSKO SAM STARTA OB PRIJAVI NA SISTEM, OB
IZSTOPU IZ MENUJEV PA SE AVTOMATSKO NAREDI ODJAVA IZ RAČUNALNIŠKEGA
SISTEMA. KER JE TERMINAL V ČASU, KO DELA PAKET PRG, KI JE BIL
STARTAN NEPOSREDNO IZ PRIJAVE NA RAČUNALNIŠKI SISTEM, POSTAVLJEN V
STATUS SLAVE (DELTA-M) OZIROMA NOCONTROL_Y (DELTA-V), UPORABNIK NE
MORE PREKINITI DELOVANJA PROGRAMSKEGA PAKETA PRG. TAKO UPORABNIK
LAHKO OPRAVLJA LE POSLE, KI SO DEFINIRANI V NJEGOVEM NABORU.

ZA STARTANJE MENUJEV NEPOSREDNO IZZA PRIJAVE NA RAČUNALNIŠKI SISTEM
JE POTREBNO NAPRAVITI NASLEDNJE:

- NA DIRECTORYJU, KAMOR SE UPORABNIK PRIJAVI, MORA OBSTAJATI .PRG
DATOTEKA, KATERE NAZIV (PRVI DEL) DOBIMO IZ DEFINICIJE PRIJAVE.

- V DEFINICIJI PRIJAVE MORA BITI PODANO:

- DELTA-M - PRIJAVA SE DEFINIRA S PROGRAMOM ACNT, V PRIJAVI MORAMO
POSEBEJ DEFINIRATI, DA NAJ BO TERMINAL SLAVE, OZNAKA
OZNAKA UPORABNIKA MORA BITI ENAKA PRVEMU DELU IMENA
UPORABNIKOVE .PRG DATOTEKE (ČE JE TO VEC, KOT 9 ZNAKOV
VELJA PRVIH 9 ZNAKOV). ZA KOMPATIBILNOST S PREJSNJIMI
VERZIJAMI JE IME .PRG DATOTEKE LAHKO TUDI PRVIH 6
ZNAKOV OZNAKE UPORABNIKA.
V LB:Š1,2CSYSLOGIN.CMD MORAJO OBSTAJATI INSTRUKCIJE,
POTREBNE ZA DELO PAKETA PRG

- DELTA-V - PRIJAVA SE DEFINIRA S PROGRAMOM AUTHORIZE, UPORABLJENI
MORATA BITI POSTAVKI /FLAGS=DISCTRL, V PRIJAVI MORA
BITI DEFINIRANA POSEBNA LOGIN PROCEDURA, V KATERI SE
NAHAJAJO POTREBNE INSTRUKCIJE ZA DELO PAKETA PRG,
GLEDE IMENA UPORABNIKOVE .PRG DATOTEKE VELJA ISTO, KOT
ZA DELTA-M (POLJE /USERNAME=). KOMPATIBILNOST ZA NAZAJ
(PRVIH 6 ZNAKOV V /OWNER=) TU NI OHRANJENA, ZATO PA NI
POTREBNO IMETI /PRIV=SYSPRV, KAR JE BILO V PREJSNJI
VERZIJI OBVEZNO.

1
2
3 4. FORSIRANJE PREKINITVE DELA
4

5 POGOSTO SE ZGODI, DA JE POTREBNO KAJ POSTORITI NA RAČUNALNIŠKEM
1 6 SISTEMU, ZA KAR JE POTREBNO NA UGAŠANJE LE-TEGA (NA PRIMER SERVIS).
7 RAČUNALNIKA SEVEDA NE SMEMO KAR UGASNITI, ČE UPORABNIKI DELAJO, SAJ
8 LAHKO TAKA NASILNA PREKINITEV MED DELOM PODRE POVEZEVE V PODATKOVNI
9 BAZI IN POVZROČI KUP TEŽAV. ZATO JE POTREBNO NAJPREJ POSKRBTI, DA
10 UPORABNIKI PRENEHAJO Z DELOM. TA POSTOPEK JE MNOGOKRAT ZAMUDEN, SAJ
11 UPORABNIKI RADI IGNORIRAJO KLICE IZ RAČUNALNIŠKEGA CENTRA IN MIRNO
2 12 DELAJO NAPREJ.

13
14 PAKET PRG NAM NUDI MOČNOST, DA S POSEBNIM UKAZOM DOLOČIMO, DA SE
15 LAHKO OBSTOJEČ PROGRAM KONČA, STARTANJE NOVIH PROGRAMOV OZIROMA
16 OBDELAV PA NI DOVOLJENO. KO DELUJOČI PROGRAM KONČA, PAKET PRG
17 NA ZASLON IZPIŠE "ŽAL TRENUTNO NE MORETE DELATI !!!" IN AVTOMATSKO
3 18 ODJAVI UPORABNIKA (SEVEDA LE, ČE SO BILI MENUJI STARTANI PREKO
19 PRIJAVE NA RAČUNALNIŠKI SISTEM).

20
21 UKAZ, S KATERIM ONEMOGOČIMO NADALJNJE DELO, SE RAZLIKUJE GLEDE NA
22 VRSTO OPERACIJSKEGA SISTEMA IN JE:

23
4 24 - DELTA-M

25
26 ASN SY:=PG77:/GBL - ZA PREPOVED NADALJNJEGA DELA

27
28 ASN =PG77:/GBL - ZA PONOVRNO DOVOLJENJE ZA DELO

29
5 30 - DELTA-V

31
32 ASSIGN/GROUP NOWORK PRG - ZA PREPOVED NADALJNJEGA DELA
33 /SYSTEM

34
35 DEASSIGN/GROUP PRG - ZA PONOVRNO DOVOLJENJE ZA DELO
6 36 /SYSTEM

37
38
39 KOT VIDIMO, JE UKAZ ZA PREPOVED DELA OBIČAJEN SISTEMSKI UKAZ. ZA
40 OPERACIJSKI SISTEM DELTA-V LAHKO ONEMOGOČIMO DELO VSEM UPORABNIKOM
11 (SISTEMSKI ASSIGN), ALI PA SAMO GRUPI UPORABNIKOV (GRUPNI ASSIGN).
7 42 NA SISTEMU DELTA-M LAHKO ONEMOGOČIMO DELO SAMO VSEM UPORABNIKOM.
43
44
45
46
47
8 48
49
50
51
52
53
9 54
55
56
57
58
59
10 60
61
62
63
64
65
11 66
67
68
69

1
2
3 5. INSTALACIJA
4

5 PROGRAMSKI PAKET PRG SE INSTALIRA S POSEBNO PROCEDURO (PRGBLD.COM ZA
1 6 DELTA-M OZIROMA PRGBLD.COM ZA DELTA-V), KI SE PO INSTALACIJI ZBRIŠE.
7 PO INSTALACIJI OSTANEJO NA RAČUNALNIŠKEM SISTEMU NASLEDNJE
8 KOMPONENTE:

9
10 - DELTA-M

11 PRG.TSK OB INSTALACIJI IZBEREMO, KAM, OBIČAJNO NA LB:Š1,55Č
2 12 PRX.TSK OB INSTALACIJI IZBEREMO, KAM, OBIČAJNO NA LB:Š1,55Č
13 PRGLOGIN.COM NA LB:Š1,2Č. Z EDITORJEM SE VNESE V SYSLOGIN.COM.
14 V STARTUP PROCEDURI JE POTREBNO DEFINIRATI INSTALIRANJE PROGRAMOV
15 PRG.TSK IN PRX.TSK.
16

17
3 18 - DELTA-V
19

20 PRG.EXE NA SYSS\$SYSTEM:
21 PRG.COM NA SYSS\$SYSTEM:
22 PRGLOGIN.COM NA SYSS\$MANAGER: TO PROCEDURO JE POTREBNO VNESTI
4 24 V LOGIN PROCEDURE UPORABNIKOV PAKETA PRG (NA KONEC)
25
26
27
28
29

5 30

31

32

33

34

35

6 36

37

38

39

40

41

7 42

43

44

45

46

8 48

49

50

51

52

53

9 54

55

56

57

58

59

10 60

61

62

63

64

65

11 66

67

68

69

M M - okolje -- sistemsko okolje

Okolje je razvito za računalnike VAX, ter deluje na OS VMS 4.x in 5.x. Organizirano je podobno operacijskemu sistemu VMS in je samostojna celota, ki je neodvisna od VMS, vendar dopolnjuje VMS pri različnih zahtevah uporabnikov računalnika.

Namen okolja:

1. pomoč na sistemskem področju (konfiguraciji računalnika, priprava računalnika za delovanje)
2. pomoč operaterjem (kontrola dela strojne opreme, kontrola programske opreme)
3. pomoč programerjem (vodenje uporabnikov pri delu, kontrola dela uporabnikov, izpis dnevnikov dela)
4. postavitve nekaterih standardov
5. poenostaviti delo v centru

Pri konfiguraciji računalnika lahko uporabljamo naslednje možnosti:

1. diski
2. tračne enote
3. terminali
4. knjižnice
5. tiskalniške vrste
6. paketne vrste
7. oblike izpisov
8. dodatne sistemske datoteke

Okolje pripravi računalnik za delo:

1. analiziranje usodnega padca sistema
2. montiranje diskov
3. nastavitve terminalov
4. postavitve tiskalniških vrst
5. postavitve paketnih vrst
6. instalacija knjižnic
7. instalacija programov
8. postavitve tabel za logična imena
9. izvajanje različnih programov in procedur

M
M
O
K
O
L
J
E

10. postavitev SWAP in PAGE datotek

Kontrola dela na računalniku:

1. prepoved dela uporabnikom
 1. več vrst prepovedi
 2. prepoved dela
 3. sporožanje prepovedi
 4. pozdravni tekst na ekranu
 5. sporočila na ekranu
 6. nasilno prekinjanje dela uporabnikom

2. delo z vrstami (tiskalnice in paketne)
 1. aktiviranje
 2. stopiranje
 3. brisanje
 4. prenašanje poslov iz ene vrste v drugo

3. pregled delovanja sistema
 1. statistični pregledi
 2. trenutno delovanje računalnika

Za končnega uporabnika okolje omogoča uporabo virtualnih terminalov, s katerimi opišemo:

1. fizični terminal
2. pripadnost organizacijski enoti
3. najbližji tiskalnik
4. seznam aplikacij, ki lahko delajo na terminalu
5. seznam organizacijskih enot
6. dodatni opisi za username (prepoved dela, potrbni diski, aplikacija, baza podatkov, prekinitvev dela z bazo, vozeli v mreži, vodenje dnevnika)
7. opisi za uporabnike v okviru username (prepoved, geslo, aplikacija, menuji, programi)
8. menuji na nivoju uporabnika (vrstni red interaktivnih programov)
9. interaktivni programi (seznam programov, način izvajanja, znižanje prioritete)

Vsi uporabniki izvajajo ob priključitvi isto proceduro.

Za hitrejše delovanja računalnika so nastavljene tabela logičnih imen, ki se postavijo ob zagonu računalnika.

Za opis in izvajanje MM okolja uporabljamo parametrsko datoteko, ki je navadna sekvenčna datoteka:

1. vsi parametri na enem mestu
2. enotnost
3. enostavna oblika
4. cluster način povezovanja računalnikov
5. razdeljena na dele
6. predpisana struktura za podatke
7. dolžina ni omejena

Podpira skoraj vse uporabniške potrebe firme, zato lahko iz VMS systemske startne procedure izpustimo (skoraj) vse; moramo pa na koncu procedure vstaviti naslednjo vrstico (brez napake):

```
$sys$sysdevice:$mm.sysmgr$sysstartup.com
```

Ob postavitvi VMS okolja se ta procedura izvede v svojem DETACH procesu in ni nevarnosti, da bi se sistem zapletel. Če uporabniki začnejo z delom pred koncem izvajanja te procedure lahko pride do nenormalnega obnašanja dela procedure, ki skrbi na nastavitve terminalov.

Okolje je razdeljeno na naslednja poglavja:

1. MM_MANAGER
2. MM_SYSTEM
3. MM_LIBRARY
4. MM_UPDATE
5. MM_PARAM
6. MM_SOURCE
7. MM_HELP
8. Priporočila
9. Spremembe in dopolnitve

4600

boot MD30\$DIA0: ↵

USER: SYSTEM
PASS: SYSTEMOPER
20PRN~~MM~~ orolje je ustavljen (SYS\$MANAGER: SYSTARTUP.V5.COM)

3100 boot dka200: ↵

1 MM_MANAGER

Na tem direktoriju so procedure, ki startajo MM OKOLJE in ga pripravijo za uporabo.

1. SETYCZO. Procedura, ki nastavi tekste v slovenščini za kontrolne tipke: čZ, čC, čY in čO.
2. STARTAPLI. Procedura za definicijo logičnih imen, ki veljajo za aplikacijo. Procedura se požene v SYSTARTUP proceduri in napravi tabelo logičnih imen z imenom LNM\$apli (primer LNM\$KADR), v kateri so shranjena logična imena, ki so definirana v MMPARAM pod labelo apliLOGIN. Procedura lahko naredi tabelo samo za eno aplikacijo, ali pa za vse.

Način klica procedure: žmmanager:startapli parameter

če hočemo samo za eno aplikacijo, napišemo ime aplikacije v parameter.

3. STARTDISK. Procedura za softversko montiranje diskov na sistemu. Najprej prebere podatke v MMPARAM datoteki pod labelo STARTDISK. Podatki morajo imeti naslednjo obliko:

```
$ ndisk      == 5!število diskov, ki jih redno montiramo
$!          diskn      fiz.ime  ime....diska  acp kap clu
$ disk1=="Dra1      DISK1          1""
$ disk2=="Dua2      DISK2          0""
```

```
ndisk      - število diskov, ki jih redno montiramo, število
ni omejeno.
fiz.ime    - fizično ime diskovne enote
ime        - oznaka labela diska
acp        - s tem določimo ACP za vsak disk posebej(ni v
uporabi)
kap        - velikost diska (ni v uporabi)
clu        - montiranje diska v CLUSTER načinu(0,1-montiranje
z /cluster)
```

4. STARTFORM. Procedura za definiranje oblik izpisov na tiskalniku. Parametri so v MMPARAM datoteki pod labelo STARTFORM in imajo naslednjo obliko:

```

$!          forma(xx)
$!          d dolžina strani
$!          d d opis forme
$sf1 =="" 1 72 234 X 12 - A4 BIANCO POKONČNI ""
$sf2 =="" 2 72 235 X 12 - IZPISKI""
$sf3 =="" 3 68 321 X 8 - A4 ZEBRA LEŽEČI ""
$sf4 =="" 4 24 SPLOŠNA POLOŽNICA ""
$sf5 =="" 5 72 380 X 12 - A3 ZEBRA ""
forma      - oznaka oblike, ki jo bo uporabljal sistem
dolžina    - dolžina strani pri izpisu (2 znaka)
opis       - opis oblike papirja

```

5. STARTGRUP. Procedura za definicijo logičnih imen, ki veljajo za grupo uporabnikov. Procedura se požene v SYSTARTUP proceduri in napravi tabelo logičnih imen na nivoju grupe in to tista imena, ki so definirana v MMPARAM pod labelo 'baza'grupa'. Procedura lahko naredi tabelo samo za eno grupo ali pa za vse. Če ne uporabljamo baze podatkov, lahko napišemo naslednjo labelo: GROUP'grupa.

Način klica procedure: žmmmanager:startgrup parameter

Če hočemo samo za eno grupo, napišemo username v parameter.

6. STARTIMAG. Procedura za instalacijo SHARE programov na sistem. Parametri so shranjeni v MMPARAM datoteki pod labelo STARTIMAG.

```

$imagfile1==""mm_system:mmuser/open/share""
$imagfile2==""reply/priv=oper!/nological_name""

```

Opisati moramo naslov programa (izvajalne kode) in opisnike za sistemski pripomoček INSTALL. Če želimo, da procedura ne naredi sistemsko logično ime z imenom programa, moramo na koncu podatkov vpisati niz '!/nological_name' (to velja predvsem za sistemske programe).

Način klica procedure: žMM_MANAGER:STARTIMAG parameter

Parameter ima lahko vrednost 'DELETE', kar pomeni da briže iz spomina instalirane programe, brez parametra pa reinstalira programe. Za vsak program naredi tudi sistemsko logično ime 'program', vrednost tega imena pa je naslov izvajalne kode na disku.

7. STARTLIBS. Procedura za instaliranje SHARE ali navadnih knjižnic. Parametri so shranjeni v MMPARAM datoteki pod labelo STARTLIBS.

```
$libsfile1===="mm_library:firmashr/open/shared/head/write"
$libsfile2===="mm_library:firmapom"
```

Opisati moramo naslov knjižice in opisnike za sistemski pripomoček INSTALL. Opisnike določimo samo za SHARE knjižnice.

Način klica procedure: %MM_MANAGER:STARTLIBS parameter

Vrednost parametra je lahko naslednja:

```
INITIALIZE - procedura nam napiše v IMAGELIB novo
SHARE knjižnico
DELETE - procedura briže SHARE knjižnico
iz spomina
''(prazno) - Procedura instalira SHARE knjižnico v
spomin, z opisniki ki so vodeni v MMPARAM datoteki. Za
vsako knjižnico naredi tudi sistemsko logično ime
'knjižnica', vrednost tega imena pa je naslov SHARE kode
na disku.
če knjižnica ni SHARE naredi samo sisemsko logično ime
LNK$LIBRARYx in kaže na izbrano knjižnico; x je
naslednja zaporedna številka začenši z 0.
```

8. STARTPRIN. Procedura za definiranje printerskih vrst na sistemu. Parametri so shranjeni v MMPARAM datoteki pod labelo STARTPRIN.

```
<ESC>4w
$! tiskalnik
$! vrsta vozle d generic
$! d d d d
$sp1===="lpa0 lpa0
$sp2===="lpb0 lpb0
$sp4===="sys#print lpa0,lpb0"
<ESC>2w
```

Opisati moramo naslednje podatke:

```
vrsta - oznaka tiskalniške vrste
vozel - Pri CLUSTER računalniku vozel, na katerem je ta
vrsta definirana
tiskalnik - fizično ime tiskalnika
generic - če je vrsta generična, seznam vrst ki jih
vključuje
ostalo - opisniki za inicializacijo tiskalniške vrste
```

Način klica procedure: ŽMM_MANAGER:STARTPRIN parameter

Vrednosti parametra so lahko naslednje:

```
INITIALIZE - inicializacija vrst v SYSTEM QUEUE
MANAGERju
DELETE      - brisanje vrst
STOP        - stopiranje vrste brez brisanja
''(prazno) - Startanje vrst s starimi podatki
```

S to proceduro lahko tudi definiramo tiskalniško vrsto, ki je obežena na HARD terminal, obvezno pa je dati v ostalo opisnik '/SPEED=nnnn', kar pomeni hitrost linije terminala.

9. STARTSORT. Procedura za definiranje logičnih imen za sortna področja. Parametri so shranjeni v MMPARAM datoteki pod labelo STARTLIBS.

```
$ define/system sortwork0 dua1:
$ define/system sortwork1 dua2:
```

Definirati moramo sistemska logična imena za sortno področje, ki kažejo na določen disk. številke tečejo od 0 do 7.

10. STARTSWAP. Procedura za definicijo in instalacijo dodatne PAGE in SWAP datoteke. Parametri so shranjeni v MMPARAM datoteki pod labelo STARTWAP.

```
$swapfile1 ==
""dua0:šmmópagefile1.sys/pagefile/size=15000""
```

Opisati moramo naslov dodatne datoteke, vrsto datoteke in velikost v bytih te datoteke. Če podana datoteka ne obstaja, naredi novo.

11. STARTTERM. Procedura za nastavitve karakteristik za terminale. Parametri so shranjeni v MMPARAM datoteki pod labelo STARTTERM.

```
<ESC>4w
```

```
$ SET TER TTA0:/PERM/VT100/SPEED=4800/PAGE=24/WIDTH=80/WRAP
$ SET TER TTA1:/PERM/VT100/SPEED=2400/PAGE=24/WIDTH=80/WRAP
$ SET TER TTA2:/PERM/VT100/SPEED=2400/PAGE=24/WIDTH=80/WRAP
$ SET TER TTA3:/PERM/VT100/SPEED=4800/PAGE=24/WIDTH=80/WRAP
$ SET TER TTA4:/PERM/VT100/SPEED=1200/PAGE=24/WIDTH=80/WRAP/MODEM/D
$ SET TER TTA5:/PERM/LA120/SPEED=4800/PAGE=64/WIDTH=132/NOBROA
$ SET TER TTA6:/PERM/VT100/SPEED=1200/PAGE=24/WIDTH=80/WRAP/FULLD/M
```

```
$ SET TER TTA7:/PERM/LA120/SPEED=4800/PAGE=64/WIDTH=132/NOBROA
<ESC>2w
```

Procedura naredi SEARCH iz MMPARAM datoteke, kjer so zapisani ukazi za nastavitve terminala. Išče naslednje nize 'SET', 'TER', '/PER' in morebitno generično ime terminalov, ki je vpisano v parameteru. Za vsak terminal, ki mu ne more nastaviti karakteristik, javi napako. Procedura, dela počasi, vendar se pri alociranem terminalu ne zaustavi; vse ostale ukaze v labeli STARTTERM v MMPARAM datoteki moramo izvesti na nek drug način, paziti moramo da ukazov v tej obliki nimamo napisanih nikjer drugje v MMPARAM datoteki.

12. STARTVRST. Procedura za definiranje paketnih vrst na računalniku. Parametri so shranjeni v MMPARAM datoteki pod labelo STARTPRIN.

```
<ESC>4w
$!                job_limit
$!                đ priority
$!      vrsta     voz     đ đ generic
$!                đ     đ     đ đ đ
$sv1 ==""slow    1 3""
$sv2 ==""fast    4 4
$sv3 ==""sys#batch 8 4
<ESC>2w
```

Opisati moramo naslednje podatke:

vrsta - oznaka tiskalniške vrste
voz - Pri CLUSTER računalniku voz, na katerem je ta vrsta definirana
joblimit - število vzporednih procesov v vrsti
priority - oznaka prioritete vrste
generic - če je vrsta generična, seznam vrst ki jih vključuje
ostalo - opisniki za inicializacijo paketne vrste

Način klica procedure: žMM_MANAGER:STARTVRST parameter

Vrednosti parametra so lahko naslednje:

```
INITIALIZE - inicializacija vrst v SYSTEM QUEUE
MANAGERju
DELETE     - brisanje vrste
STOP       - stopiranje vrste brez brisanja
''(prazno) - Startanje vrst z starimi podatki
```

13. SYSTARTUP. Procedura za postavitev MM OKOLJA pri startanju računalnika. Način klica procedure:

```
##SYS$DEVICE:##MM.SYSMGR&SYSTARTUP ali ##MM_MANAGER:SYSTARTUP
```

Procedura sama ugotovi, kdaj in kako je bila startana (pri postavitvi sistema ali interaktivno) in se ustrezno izvaja.

Najprej testira, če obstaja datoteka SYS\$SYSTEM:SYSDUMP.OLD; to mu pomeni, da mora narediti dump spomina od zadnjega padca sistema. Zato morajo biti nastavljeni pravilno tudi sistemski parametri (SAVEDUMP, DUMPBUG). Datoteko lahko pregledujemo z ukazom ANALYZE/CRASHDUMP 'datoteka'. Ko se sistem uravnovesi moramo pobrisati to datoteko.

Najprej definira vsa logična imena za MM OKOLJE:

```
MM__SYSTEM
MM__MANAGER
MM__SOURCE
MM__LIBRARY
MM__HELP
MM__PARAM
MM__SPECIFIC
MM__UPDATE
MM__OPER
```

Tu procedura konča, če je vrednost "MIN" v sistemskem parametru STARTUP_P8. To pomeni, da se postavijo samo logična imena MM OKOLJA, vse ostalo pa moramo naložiti ročno.

Nato izvede vse definicije, ki so naštetje v MMPARAM datoteki pod labelo DEF_DEFINICIJ.

Potem prepove delo vsem uporabnikom na sistemu, razen za SISTEM in OPER.

Pobriše stare OPERATOR.LOG (keep=3).

Nato izvaja procedure, ki so v seznamu v MMPARAM datoteki pod labelo DEF_PROCEDUR. Zadnje #tevilu izvedene procedure je #tevilu, ki je definirano v simbolnem imenu STEVILO_PROCEDUR.

Pravilno moramo določiti seznam definicij in seznam procedur, da se bo MM OKOLJE pravilno postavilo. Primer kaže ta dva seznama:

```
#!seznam definicij
# definicija1 == "def_virter"
# definicija2 == "def_debugger"
```



```

$ definicija3 == "def_diski"
$ definicija4 == "def_trakovi"
$ definicija5 == "def_firma_naziv"
$ definicija6 == "def_announce"
$ definicija7 == "def_razne"

#! seznam procedur pri postavitvi MM OKOLJA
$ stevilo_procedur==14
$ procedura1 == ""$mm_manager:startdisk""
$ procedura2 == ""$mm_manager:startterm""
$ procedura3 == ""$mm_manager:startimag""
$ procedura4 == ""$mm_manager:startlibs""
$ procedura5 == ""$mm_manager:startsort""
$ procedura6 == ""$mm_manager:startform""
$ procedura7 == ""$mm_manager:startvrst""
$ procedura10== ""$mm_manager:startswap""
$ procedura11== ""$mm_manager:startgrup""
$ procedura12== ""$mm_manager:startapli""
$ p2:="izvedi drugo proceduro"

```

14. MMLOGIN. Vključitvena procedura za vse uporabnike, ki uporabljajo MM OKOLJE. Klic te procedure moramo vpisati v AUTHORIZE sistemskem pripomočku pri vsakem uporabniku z opisnikom '/LGICMD=MM_MANAGER:MMLOGIN'.

Procedura najprej požene program MMUSER in na podlagi statusa programa nadaljuje. Če je aplikativni uporabnik, gre na menuje, drugače nadaljuje na podlagi statusa:

```

status = 11      izvede LOGIN.COM in konca status = 3
                naredi SET HOST 'vozel' in nato STOP RUN

```

Pri menujih naredi nekaj DEASSIGNov in preveri status izvajanja programa, ter ga obdela.

Procedura tudi opiše, kakšen naj bo PROMPT na nivoju DCL ukazov. Privzeta vrednost je naslednja: '\$' plus 'R', če dela uporabnik v REMOTE načinu, 'x' in oznaka računalnika.

15. MODPAR. Procedura za nastavljanje sistemskih parametrov. Vsako spremembo napiše v datoteko MODPARAMS.DAT, kar služi pri spreminjanju verzij operacijskega sistema. Možno je pisati tudi komentarje.

2 MM_SYSTEM

Na tem direktoriju so vsi programi in procedure, ki jih uporabljamo za tekoče delo v mm okolju.

1. AZUMEN.COM - procedura za izpis MMOKOLJA za vse uporabnike: Izpiše userje, uporabnike, menuje in programe; če ugotovi logične napake, jih napiše na listo
2. AZUMEN.EXE - program za ažuriranje datotek USERJI na MM_PARAM direktoriju, UPORAB, MENUJI, PROGRAM na SYS\$LOGIN direktoriju
3. AZURIND.COM - procedura za ažuriranje indeksnih datotek. Podatki so shranjeni ali bodo shranjeni na sekvenčni datoteki.
4. BRIEKR.COM - brisanje ekrana in nastavitve na 80-132
5. BRIPOS.COM - rokovanje z vrstami na sistemu
6. CISTIX.COM - brisanje nepotrebnih datotek aplikacije
7. COMLIN.CLD - CLD ukazna procedura za prevajalnik
8. COMLIN.COM - procedura za prevajalnik
9. COMLIN.EXE - program za prevajalnik
10. EDITOR.COM - procedura za editiranje cobolskih programov
11. INDAZU.COM - procedura za ažuriranje indeksnih datotek
12. IZTRAK.EXE - program za prepis iz trakov
13. LIBRAR.COM - procedura za knjiznice
14. LOGING.EXE - program za vodenje dnevnika dela
15. LOGIZP.EXE - program za izpis dnevnika dela
16. MMEDT.COM - procedura za editiranje, prevajanje in povezovanje COBOLskih programov
17. MMEDT.EDT - ukazna procedura za EDT editor (predprocedura)
18. MMPON.COM - procedura za ponavljanje DCL ukazov
19. MMSOR.COM - procedura za sortiranje
20. MMUAF1.EXE - program za izpis prepovedi dela
21. MMUAF2.EXE - program za izpis podatkov o userjih in uporabnikih: NOCAPTIVE uporabniki, DISCTLY, uporabniki, ki nimajo MMLOGIN, in vseh uporabnikov, ki pridejo na \$ znak.
22. MMUAF3.EXE - program za prenos zapisa iz UAF v globalni simbol
23. MMUSER.EXE - program za vključitvene menuje
24. MMUSE1.EXE - program za spreminjanje gesel uporabnikov
25. MMUSE9.EXE - program za kontrolo gesel po datumih
26. MMUSE9.COM - procedura za kontrolo gesel po datumih
27. MMMENU.EXE - program za menuje
28. NATRAK.EXE - program za pisanje na trakove

29. RUNOLP.EXE - Program za prevod RUNOFF datotek za izpis
30. SETDEF.COM - procedura za SET DEFAULT ukaz
31. STRING.COM - procedura za STRING ali UNSTRING
32. TRUREC.EXE - program za krajšanje zapisov do določenega niza
33. UPORA1.EXE - program za preminjanje gesla upravnika
34. VIRTER.EXE - program z azuriranje virtualnih terminalov
35. ZAMENJAJ.COM-procedura za zamenjavo nizov v datotekah

V nadaljevanju so navodila za uporabo MM sistemskih programov. Navodil ni za enostavne programe ali procedure, ter za tiste programe, ki nimajo parametrov. Vsi programi in procedure imajo kratko opisano delovanje in navodila za uporabo v samem izvornem modulu.

2.1 AZUMEN

Program za pripravo okolja uporabnikov za izvajanje programov. Program uporablja naslednje datoteke:

- | | | |
|----|---------|---------------------------|
| 1. | USERJI | na MM_PARAM:USERJI.IND |
| 2. | UPORAB | na SYS\$LOGIN:UPORAB.IND |
| 3. | MENUJI | na SYS\$LOGIN:MENUJI.IND |
| 4. | PROGRAM | na SYS\$LOGIN:PROGRAM.IND |

Na začetku program preveri UIC uporabnika, in če je ta manjši od 11 vpraša za disk in direktorij, kjer naj vzame te datoteke (vrednost SYS\$LOGIN).

Program ažurira USERNAME, uporabnike, menuje in programe samo za MM OKOLJE. Za ažuriranje uporabnikov v SYSUAF moramo uporabljati sistemski pripomoček AUTHORIZE.

V okolju lahko izbiramo, ali bo imel Username podrejene uporabnike ali pa bo imel samo enega (ključ v UPORAB je sestavljen samo iz USERNAME in space). Podpira tudi izvajanje samo enega programa ali procedure pri posameznem uporabniku. Ta program ali proceduro vpišemo v polje DEFDIR v datoteki UPORAB (za izvajanje programa moramo vpisati RUN)

Datoteka USERJI ima naslednjo strukturo zapisa:

- | | | | |
|-----|---------------------|---|-------------------------------------|
| 1. | username | - | sistemska ime uporabnika |
| 2. | indikator | - | indikator aktivnosti uporabnika |
| 3. | PE | - | pripadnost poslovni enoti |
| 4. | naziv | - | opis uporabnika |
| 5. | vozel | - | ime vozla za delo uporabnika |
| 6. | diski | - | seznam potrebnih diskov |
| 7. | baza | - | ime baze podatkov |
| 8. | bazadol
delu | - | deaktiviranje baze po končanem delu |
| 9. | datpar | - | naslov DATPARjev |
| 10. | seznam | - | naslov seznama |
| 11. | opisniki
spodaj) | - | 10 opisnikov za userja (glej |

Polje opisnikov je razdeljeno na deset znakov:

1 znak - N = Nima podrejenih uporabnikov, vendar mora v datoteki uporabnikov obstajati zapis s praznim uporabnikom

2 znak - N = Ne piše v LOGINGFILE

3-9 ostali znaki so rezervirani za bodočnost

Zapis UPORAB

Datoteka UPORAB ima naslednjo strukturo zapisa:

- | | | | |
|-----|---------------------|---|----------------------------------|
| 1. | username | - | sistemska ime uporabnika |
| 2. | uporabnik | - | uporabnik |
| 3. | indikator | - | indikator aktivnosti uporabnika |
| 4. | pe | - | pripadnost poslovni enoti |
| 5. | naziv | - | opis uporabnika |
| 6. | telefon | - | telefon |
| 7. | geslo | - | geslo uporabnika |
| 8. | menu | - | navjizji nivo menujev |
| 9. | ddir | - | naslov direktorija ali procedure |
| 10. | datpar | - | naslov DATPARjev |
| 11. | seznam | - | naslov seznama |
| 12. | datspr | - | datum zadnje spremembe |
| 13. | opisniki
spodaj) | - | 10 opisnikov za uporabnika (glej |

Polje opisnikov je razdeljeno na deset znakov:

1 znak - E = nima menujev, ampak izvaja samo Eno proceduro ali program

2-9 ostali znaki so rezervirani za bodočnost

Podatki pri PROGRAM

1. program - oznaka programa
2. simbol - logična oznaka programa za uporabnike
3. opis - opis programa
4. izvajanje - način izvajanja programa; če je to grupni nivo menuja, potem je tukaj vrednost 'P'
5. prioriteta - znižanje prioritete pri izvajanju programa
6. prepoved - prepoved izvajanja programa ali podmenuja

Delovanje programa:

Najprej vnesemo Username; če je PF1 do PF4 program postavi v način ažuriranje programov. Return pri Username pa pomeni izpis vseh username na sistemu (iz datoteki USERJI). Pri izbranem Username ga lahko dodajamo ali spreminjamo. Na koncu ažuriranja gremo s tipko RETURN naprej na del za uporabnike. Return tipka izpiše vse uporabnike v okviru enega username.

Pri izbranem Uporabniku ga lahko dodajamo ali spreminjamo. Na koncu ažuriranje gremo s tipko RETURN naprej na del za menuje. Tu lahko dodajamo, brižemo ali spreminjamo menuje. Spremenimo lahko tudi indikator menuja. Brisanje menuja ne pomeni tudi brisanje ekrana. Pri vsaki akciji morame vnesti tudi številko vrstice v menujih; zaveznost ni zhtevana je pa priporočljiva. Ni možnosti vrivanja menuja med dva že ostoječa menuja.

Program izpisuje USERJE UPORABnike MENUJE in PROGRAMe na SYS\$OUTPUT. Opozori tudi na razne napake, če jih seveda odkrije. Način klica programa: '\$MMSYSTEM:AZUMEN IZPIS'. Program dela za MM_OKOLJE, ki je napisano na SYS\$LOGIN področju. Za izpis vseh podatkov na sistemu je narejena procedura AZUMEN na MM_SYSTEM. Na DEFAULT direktorij napiše datoteko AZUMEN.LIS.

2.2 Program MMUSER.

Program zamenjuje programa USERJI in UPORAB. Način dela je naslednji: če je UIC grupa < 11 definira samo globalne simbole za uporabnika (navadno SYSTEM in OPER). Če username ni ključ v USERJI.IND, pregleda VIRTER, naredi globalne simbole in požene LOGIN.COM na SYS\$LOGIN direktoriju. Če je to aplikativni uporabnik, naredi naslednje: pregleda virter, naloži bazo podatkov, preveri USERJI in UPORAB, nastavi logična imena in glabalne simbole, ter pripravi za delo s programom MMMENU.

2.3 Program MMMENU

Program vodi uporabnika po menujih. Program prebere podatke iz UPORAB.IND, MENUJI.IND in PROGRAM.IND, jih preuredi in jih da na voljo uporabniku. Ko uporabnik izbere možnost, program pripravi ukaz v simbol: 'IZVEDI'. Po končanem izvajanju preveri uspešnost izvajanja in gre ponovno na isti meni, kjer je uporabnik startal ukaz.

2.4 NAPAKE pri MMUSER MMMENU

```
"%USS-F-Napaka na datoteki USERJI.IND"
    Napaka na datoteki, glej dodatni opis na terminalu
    (protekcija,zaklenjena).
"%USS-F-Username ima prepoved za delo!!!"
    prepoved dela za USERNAME; indikator v datoteki USERJI.IND
    ni enako "01".
"%USS-F-Vaši podatki še niso pripravljeni"
    Podatki na spominskih medijih niso pripravljeni.
    Vstavi previlni disk, ali pa popravi seznam diskov USERJIh.
"%USS-F-Ne moreš uporabljati " ussaplik
    Na tem terminalu ne moreš uporabljati aplikacije.
    Popravi VIRTER.
"%USS-F-Baza podatkov ni pripravljena"
    Bazo podatkov se ne da pravilno naložiti. Preveri
    napake v log datoteki baze podatkov.
"%USS-F-Skrbnik prepovedal delo"
    Prepoved dela za uporabnike aplikacije z logicnim imenom
    PREPOVED, ki na grupnem nivoju, ali celo v tabeli
    logicnih imen za aplikacijo.

"%VIR-F-Napaka na datoteki VIRTER_FILE"
    Napaka na datoteki, glej dodatni opis na terminalu
    (protekcija,zaklenjena).
"%VIR-F-Ta terminal ni predviden za delo"
    Terminal ni naveden v VIRTERju. Popravi VIRTER.
"%VIR-W-Ne moreš uporabljati tiskalnika "
    Nepravilno definicija za tiskalnik, ki ga uporablja
    XEKRAN za prerise ekranov.

"%UPO-F-Napaka na datoteki UPORAB.IND"
    Napaka na datoteki, glej dodatni opis na terminalu
    (protekcija,zaklenjena).
```

"%UPO-W-Nepravilna oznaka uporabnika"
 Oznaka uporabnika je nepravilna.

"%UPO-F-Preveckrat vpisan napacen tekst"
 Napačni tekst je bil 5 vpisan narobe. Sistem avtomatsko izkljuci terminal.

"%UPO-F-Prepoved dela za uporabnika"
 Prepoved dela za Uporabnika; indikator v datoteki UPORAB.IND ni enako "01".

"%UPO-F-Manjka memu za uporabnika"
 Menuji niso logicno pravilno povezani. Preveri datoteke UPORAB.IND in MENUJI.IND. Za definicijo okolja uporabljaj program AZUMEN.

"%SYS-F-prepoved dela za uporabnike"
 Prepoved dela za uporabnike z logicnim imenom PREPOVED, ki je lahko na sistemskem ali grupnem nivoju, ali celo v tabeli logicnih imen za aplikacijo.

"%PRO-F-Napaka na datoteki PROGRAM.IND"
 Napaka na datoteki, glej dodatni opis na terminalu (protekcija,zaklenjena).

"%PRO-F-Nepravilna oznaka programa"
 Pri izbiri programov je vpisana oznaka programa, ki ga ni na seznamu.

"%MEN-F-Napaka na datoteki MENUJI.IND"
 Napaka na datoteki, glej dodatni opis na terminalu (protekcija,zaklenjena).

"%MEN-F-Nepravilna oznaka menuja"
 Menuji niso logicno pravilno povezani. Preveri datoteke UPORAB.IND in MENUJI.IND. Za definicijo okolja uporabljaj program AZUMEN.

2.5 BRIPOS

Procedura za pregled in kontrolo BATCH in PRINT vrst na sistemu

Kličemo jo s simbolom "Q", hkrati z ustreznimi parametri. Če ne navedemo nobenega parametra potem na ekran dobimo kratko informacijo o vseh poslih v vrstah (BATCH + PRINT) !

Prvi parameter je lahko :

1. B - pregled BATCH vrst
2. P - pregled PRINT vrst (linijski tiskalniki)
3. N - nastavitev forme za tiskalnike

Drugi parameter omogoči še ožji izbor prvega parametra :

1. n - zaporedna številka BATCH ali PRINT vrste
2. - izpis vseh

Tretji parameter določa glede na izbor prvih dveh parametrov še sledeče možnosti :

1. nnnn - številka posla (ENTRY), ki ga želimo brisati
2. ff - številka forme za izbran tiskalnik
3. F - podroben pregled izbrane vrste (FULL)

Četrty parameter je mogoč le v primeru spremembe forme za tiskalnike in se omogoča zakasnitev nastavitve za 15 sekund (uporabno v primeru, da je izbrani tiskalnik v stanju ON-LINE). Parameter vpišemo v obliki "W".

Po vsaki prvi priključitvi uporabnika na sistem procedura dela nekaj sekund dlje, ker si priredi nekatere simbole za izvajanje. Opisi nastavitvev so v dokumentaciji MMOKOLJE.MEM (poglavje STARTFORM in STARTVRST), dejanske nastavitve pa so v proceduri MMPARAM.COM.

2.6 MMUSE9

Program in procedura za kontrolo spremembe gesla pri uporabnikih

Program MMUSE9 pa pregleduje spremembe gesel. Če geslo uporabnika ni spremenjeno več kot 120 dni (4 meseca) program generira novo geslo. Za SKRBNIKA pa vzame geslo iz GET_FOREIGN polja, če je pa to polje prazno vzame vrednost logičnega imena

"firma_NAZIV_1".

Procedura MMUSE9 pa poganja program MMUSE9 za vse ali samo za enega uporabnika. Prvi parameter je ime uporabnika, drugi parameter pa je vrednost gesla za SKRBNIKE.

2.7

Program MMUSE1 naj uporabljajo skrbniki aplikacij za spreminjanje gesel uporabnikov. Spreminjanje je interaktivno. Geslo morajo spremeniti pri vsakem uporabniku posebej.

2.8 VIRTER

Program za ažuriranje virtualnih terminalov. Uporablja datoteko VIRTER.IND na MM_SPECIFIC direktoriju. Opišemo naslednje podatke:

- | | | |
|----|------------------------|---|
| 1. | virtualna oznaka | - virtualna oznaka terminala VTnnn |
| 2. | vrata
računalniku | - dejanska oznaka vrat na |
| 3. | indikator | - indikator aktivnosti terminala |
| 4. | tip
Printer, Remote | - tip terminala: Terminal, |
| 5. | šifra OE
SIFPE.PAR | - oznaka poslovne enote iz datoteke |
| 6. | lokacija | - lokacija terminala |
| 7. | telefon | - najbližji telefon pri terminalu |
| 8. | tiskalnik
prelise | - virtualna oznaka tiskalnika za |
| 9. | aplikacije | - dovoljenje za delo v aplikaciji;
D-lahko uporabljamo aplikacijo na terminalu, ostalo - ne
moremo je uporabljati |

Definirati moramo tudi REMOTE terminale (RTA*****), terminale priključene na TERMINAL SERVER ter terminale na INFO mreži in DIAL-UP terminale. Oblika ključa je naslednja: xxx*****, kjer

xxx pomeni prvih nekaj alfa znakov fizičnega imena terminala.

3 MM_LIBRARY

Na direktoriju MM_LIBRARY so shranjene datoteke, ki se uporabljajo pri kreiranju knjižnic in same knjižnice. Vsaka mm lahko dodaja svoje knjižnice, tako SHARE kot tudi OLB knjižnice.

Trenutno so naslednje datoteke:

1. FIRMAOLB.OLB - OLB knjižnica
2. FIRMALIB.TLB - tekstovna knjižnica za definicijo zapisov
3. FIRMANAV.TLB - tekstovna knjižnica za navopdila za izvajanje ali klicanje programov in podprogramov

4 MM_UPDATE

Na tem diraktoriju so datoteke, ki jih rabimo za distribucijo in za pripravo le-te.

1. MMINSTAL.COM - procedura za kreiranje distirbucije in samo distribucijo
2. MMINSTAL.RNO - opis delovanje procedure MMINSTAL
3. MMOKOLJE.RNO - opis MM OKOLJA

5 MM_PARAM

Vsi podatki, ki jih rabimo za opis MM OKOLJA, opis uporabnikov, opis samega računalnika so shranjeni v datoteki MMPARAM. V bistvu je to procedura, ki lahko kliče še podrejeno proceduro, če računalnik dela v CLUSTER načinu.

MMPARAM datoteka je razdeljena na labele; začetna labela je kar labela, končna pa je sestavljena iz labele in 'EXIT'. Glej primer. Prva labela je 'START_MMPARAM' in je obvezna!!!!. V vsaki labeli skočimo (ukaz GOTO) na konec, da poženemo še podrejeno MMPARAM proceduro, če je potrebno. To definiramo s parametrom p2, ki mu damo vrednost različno od SPACE (p2="poženi se podrejeno proceduro").

Prireditvev simbolnih imen je predpisana in ima obliko:

```
ime_simbola == ""vrednost simbola ""
```

To mora biti globalno simbolne ime, vrednost imena pa mora biti omejena na vsaki strani s tremi dvojnimi narekovaji!!!!!!

Podrejena procedura je na direktoriju MM_PARAM:š.'vozel'ć, kjer je vozel ime vozla v CLUSTERju.

Tu opisujemo grupo uporabnikov, aplikacijo, MM OKOLJE, sistemsko okolje..

Direktorij MM_SPECIFIC je močno povezan s direktorijem MM_PARAM. Na tem direktoriju so podatki, ki natančno opisujejo vozel v CLUSTER načinu povezovanja računalnikov (pri homogenem načinu povezovanja).

Na MM_PARAM direktoriju so naslednje datoteke:

- | | | | |
|----|-------------|---|---------------------------------|
| 1. | APLIK.PAR | - | seznam aplikacij |
| 2. | SIFPE.PAR | - | seznam poslovnih enot |
| 3. | MMPARAM.COM | - | parameterska datoteka |
| 4. | USERJI.IND | - | seznam aplikativnih uporabnikov |
| 5. | VIRTER.IND | - | tabela virtualnih terminalov |

V nadaljevanju je posebej opisana samo datoteka MMPARAM.COM; Ostale datoteke pa so vpisane v priročnikih za uporabo programov.

1. Okolje grupe uporabnikov. To okolje velja za vse uporabnike v okviru ene grupe. Ponavadi so to uporabniki, ki delajo v isti bazi podatkov. Zato je tudi oznaka labela sestavljena iz oznake področja baze podatkov in trimestne številke grupe. Primer:

```
<ESC>§4w
$APLIKA_210: !===== logična imena za bazo APL
$ DEFINE/GROUP APLI_MAS DS1:§2106
$ DEFINE/GROUP APLI_VAR DS2:§2106
$ DEFINE/GROUP APLI_VAR1 DS2:§2106
$ DEFINE/GROUP APLI_DAT DS2:§APLI.DAT6
$ DEFINE/GROUP APLI_EXE DSO:§APLI.EXE6
$ DEFINE/GROUP APLI_COM DSO:§APLI.COM6
$ goto exit
$APLIKA_210_exit:
<ESC>§2w
```

Zaradi hitrosti izvajanja testiramo, če je bil ta del procedure že kdaj klican, saj vse velja na nivoju grupe. Definiramo vsa skupna logična imena, ki jih bodo uporabljale vse aplikacije v grupi.

Postopek ponovimo za vsako grupo podatkov posebej.

2. Okolje aplikacije. Pri tem opišemo okolje za samo aplikacijo. Vse definicije veljajo na procesnem nivoju.

```
$APLIlogin: !=====logična imena za aplikacijo APLI
$ define APLI_ext ds2:§APLI.pom6
$ define APLI_tmp ds2:§APLI.pom6
$ define APLI_1st ds2:§APLI.1st6
$ define APLI_cob ds0:§APLI.cob6
$ define APLI_pom ds0:§APLI.pom6
$ define APLI_lib ds0:§APLI.lib6
$ define APLI_log ds0:§APLI.log6
$ define APLI_dok ds0:§APLI.dok6
$ goto exit
$APLIlogin_exit:
```

3. Logična imena za diske. Vsakemu fizičnemu disku priredimo enega ali več logičnih imen, naj bi imeli naslednjo sktrukturo: DSx, kjer je x zaporedna številka. S tem naj bi ločili zaključene dele podatkov na računalniku, obenem pa pri uporabi teh logičnih imen za diske ni problemov pri prenosu aplikacij.

Primer:

```

<ESC>§4w
$def_diski: !=====logična imena za diske
$   define/system   ds0           'f$logical("sys$sysdevice
$   define/system   ds1           ___$1$dual:
$   define/system   ds3           ___$1$dual:
$   define/system   ds2           ___$1$dual:
$   define/system   ds5           ___$1$dual:
$!  def/sys/exe/tran=(term,conc) ds3 $1$dual:šdisk3.6
$   goto exit
$def_diski_exit:
<ESC>§2w

```

4. Logična imena trakov. Tračnim enotam priredimo logična imena standardne oblike (MTx), zaradi krajšega pisanja ali poenotene uporabe neglede na fizično ime enote.

Primer:

```

$def_trakovi: !=====logična imena za tražne enote
$   define/system   mt0           mua0:
$   define/system   mt1           mub0:
$   goto exit
$def_trakovi_exit:

```

5. Logično ime za spročilo. Tukaj definiramo logično ime SYS\$ANNOUNCE in SYS\$WELCOME, katere sistem rabi pri priključitvi uporabnika.

SYS\$ANNOUNCE sistem napiše na terminal, takoj ko uporabnik pritisne prvo tipko na terminalu, SYS\$WELCOME pa napaiše ko se je uporabnik uspešno priključil na sistem.

Primer:.

```

<ESC>§4w
$def_announce: !=====definicija announce in wellcomm
$ define/sys/exe sys$announce      "žmm_specific:announce.tx
$ define/sys/exe sys$welcome "Uspešno delo na sistemu ''vozel'
$   goto exit
$def_announce_exit:
<ESC>§2w

```

6. Nazivi firme. Opišemo tri nazive firm: kratki, srednji in dolgi. Samo logično ime pove za kateri naziv gre. Imena različno uprabljamo, npr. v interaktivnih programih uporabljamo kratki naziv, pri izpisnih listah dolgi naziv...

Primer:

```
<ESC>§4w
$def_firma_naziv:!=====definicija nazivov za nasl
$   define/system   firma_naziv_1   ""
$   define/system   firma_naziv_2   "   "
$   define/system   firma_naziv_3   "   "
$   goto exit
$def_firma_naziv_exit:
<ESC>§2w
```

7. Okolje prevajalnika. Ta prevajalnik smo naredili zaradi zaporednega prevajanja in povezovanja programov. Izvaja se v paketni vrsti (SLOW), ki ima JOBLIMIT=1. Pri tem ne obremenjujemo sistema. Definirati moramo DCL ukaz PREVEDI s sistemskim ukazom SET COMMAND COMLIN. Tega shranimo v sistemsko tabelo DCL ukazov (DCLTABLES.EXE) že pri instalaciji MM OKOLJA. Ukaz PREVEDI požene program COMLIN, ta pa da prevajati program v paketno vrsto. Prevajalnik zna prevajati vse vrste programov(?!).

Opisniki pri ukazu PREVEDI:

```
<ESC>§4w
parameter   ime-programa-za-prevajanje
qualifier   šnočCOMPILE           -prevajanje
qualifier   šnočLINK              -povezovanje
qualifier   LIST                  -listing programa
qualifier   MAP                   -map programa
qualifier   DEBUG                 -prevajanje z DEB
qualifier   LANGUAGE (DEFAULT="COBOL") -jezik
qualifier   DELIMITER, (DEFAULT="""$ """) -inline komentarj
qualifier   EXECUTE               -izvajalna koda
qualifier   LIBRARY               -imena dodatnih k
qualifier   OPTIONS_LIBRARY       -imena opt datote
qualifier   QUEUE, (DEFAULT="SLOW") -ime vrste
qualifier   CONDITIONALS, (DEFAULT="S") -jezikovno področ
<ESC>§2w
```

8. Opis knjižnic. MM OKOLJE podpira SHARE knjižnice in OLB knjižnice. Če je SHARE knjižnica mora biti napisan obvezno opisnik '/SHARE'. Primer:

```
<ESC>§4w
$startlibs:!=====definicije knjižnic in instalacija
$   libsfile1==""mm_library:firmashr/open/shared/head/write"
$   libsfile2==""mm_library:firmapom""
```



```
$ goto exit
$startlibs_exit:
<ESC>š2w
```

9. Opis programa za instaliranje. Če nek program uporablja več uporabnikov naenkrat, lahko ta program instaliramo v spomin. Na ta način prihranimo nekaj spomina za same kode programov. Napičemo lahko vse opisnike ki jih omogoča sistemski INSATALL pripomoček. Primer:

```
$startimag: !=====instaliranje programov
$ imagfile1====""ds0:šAPLI.exečapli01.exe/open/share""
$ imagfile2====""sys$system:reply/priv=oper!/nological_name""
$ goto exit
$startimag_exit:
```

10. Opis montiranja diskovnih enot. Napisati moramo seznam diskovnih enot, ki jih je potrebno montirati pri startu računalnika. Glej primer pri opisu procedure STARTDISK.
11. Opis karakteristik terminalov. Opišemo vse karakteristike terminalov, ki jih rabimo za tekoče delo. Glej primer pri opisu procedure STARTTERM.
12. Opis paketnih vrst. Opišemo vse paketne vrste, ki jih bomo uporabljali na računalniku ali na več računalnikih v CLUSTRU (če je CLUSTER homogen). Najprej opišemo posamezne vrste, nazadnje generične. QUEUE MANAGER mora biti predhodno startan!! Glej primer pri opisu procedure STARTVRST. Pod ostalo lahko napišemo vse opisnike za startanje vrst.
13. Opis tiskalniških vrst. Opišemo vse tiskalniške vrste, ki jih bomo uporabljali na računalniku ali na večih računalnikih v CLUSTRU (če je CLUSTER homogen). Najprej opišemo posamezne vrste, nazadnje generične. QUEUE MANAGER mora biti predhodno startan!! Glej primer pri opisu procedure STARTPRIN. Pod ostalo lahko napišemo vse opisnike za startanje vrst.
14. Opis oblike izpisov. Opišemo oblike (forme) papirja, na katerega bomo tiskali izhodne liste. Glej primer pri opisu procedure STARTFORM.

15. Opis debuggerja. Opisati moramo logično ime `DBG$INPUT` in ima kar stalno vrednost. Primer:

```
$def_debugger: !=====definicija debuggerja
$   define/system   dbg$input       sys$command
$   goto exit
$def_debugger_exit:
```

16. Opis tabele virtualnih terminalov. Opisati moramo naslov tabele virtualnih terminalov. Stalna vrednost naslova ja na `MM_SPECIFIC`, zaradi `CLUSTER` povezovanja računalnikov. Primer:

```
$def_virter: !=====definicija virterja
$   define/system   virter_file     mm_specific:virter.ind
$   goto exit
$def_virter_exit:
```

17. Opis poslovnih enot. Opisati moramo vse poslovne enote, ki so v okviru temeljne firme, kjer je instaliran računalnik. Primer:

```
$def_poslovnih_enot: !=====definicija poslovnih enot
$!1 znak           zaporedna oznaka
$!1 znak           prazno polje
$!4 znaki          oznaka poslovne enote
$!1 znak           prazno polje
$!x znakov         ime poslovne enote
$   create mm_param:sifpe.par
1 1234 firma
$   goto exit
$def_poslovnih_enot_exit:
```

18. Opis seznama aplikacij. Tu so napisane vse aplikacije, ki jih uporabljajo uporabniki. Povezava je z `VIRTER`jem; vsak uporabnik mora imeti dovoljenje za delo z določeno aplikacijo in na določenem terminalu. Primer:

```
$def_aplikacij: !=====definicija aplikacij
$!4 znaki          oznaka aplikacije
$!3 znaki          vrednost ' - '
$!x znakov         opis aplikacije
$   create mm_param:aplik.par
APLI -             Knjigovodski informacisjki sistem
$   goto exit
$def_aplikacij_exit:
```

19. Opis seznama procedur za izvajanje pri startu MM OKOLJA. Napišemo seznam vseh procedur, ki jih moramo izvesti pri startu MM OKOLJA. število teh procedur omejuje simbolno ime STEVILO_PROCEDUR. Primer:

```
$def_procedur: !=====definicije procedur
$   stevilo_procedur==14
$   procedura1 == ""%mm_manager:startdisk""
$   procedura2 == ""%mm_manager:startterm""
$   procedura3 == ""%mm_manager:startimag""
$   procedura4 == ""%mm_manager:startlibs""
$   procedura5 == ""%mm_manager:startsort""
$   procedura6 == ""%mm_manager:startform""
$   procedura7 == ""%mm_manager:startvrst""
$   procedura10== ""%mm_manager:startswap""
$   procedura11== ""%mm_manager:startgrup""
$   procedura12== ""%mm_manager:startapli""
$   p2="izvedi drugo proceduro"
$   goto exit
$def_procedur_exit:
```

20. Opis raznih definicij. Opišemo ime računalnika in vrednost prompta. Podane so nekatere rešitve, lahko pa jih tudi spreminjamo. Narediti je potrebno logični imeni IME_RACUNALNIKA in PROMPT_STRING. Primer:

```
<ESC>%4w
$def_razne: !===== definicije raznih definicij
$!   definicija imena računalnika:
$!       lahko definiramo svoje ime
$!       če je del CLUSTER sistema je ime kar ime vozla v
$   ime_racunalnika="racunalnik"
$vmc4   if f$getsyi("CLUSTER_MEMBER") then ime_racunalnik
$   define/system ime_racunalnika 'ime_racunalnika
$!
$!   prompt dela samo v verziji 4.x in naprej
$!   vzamem prvi znak imena racunalnika(ponavadi)
$   define/system prompt_string 'f$extrcat(0,1,f$logical("ime
$   goto exit
$def_razne_exit:
<ESC>%2w
```

21. Opis definicij. Seznam definicij, ki jih rabimo pri postavitvi MM OKOLJA. Pri postavitvi se najprej izvedejo te definicije. Seznam mora imeti zvezne številke simbolnih imen!! Primer:

```
$def_definicije: !=====   razne definicije
$   definicija1 == "def_virter"
```

```

$   definicija2 == "def_debugger"
$   definicija3 == "def_diski"
$   definicija4 == "def_trakovi"
$   definicija5 == "def_firma_naziv"
$   definicija6 == "def_announce"
$   definicija7 == "def_razne"
$   goto exit
$def_definicij_exit:

```

22. Okolje za sortiranje. Opis pomožnih datotek za sortiranje. Primer:

```

$startsort: !=====logična imena za sortworke
$   define/system   sortwork0 ds3:
$   define/system   sortwork1 ds3:
$   define/system   sortwork2 ds3:
$   goto exit
$startsort_exit:

```

23. Opis SWAP in PAGE datoteke. Opis dodatne datoteke za SWAP in PAGE. primer:

```

$startswap: !=====startanje druge page_file in swapfile
$swapfile1 ==""ds1:šmmćpagefile1.sys/pagefile/size=4096""
$   goto exit
$startswap_exit:

```

6 MM_SOURCE

Na tem direktoriju so vse izvirne verzije programov in podprogramov, ki jih uporablja MM OKOLJE. Tu so tudi vsi aplikativni programi, ki sestavljajo knjižnice. Za vsak program (?) je tudi datoteka za opis in navodila za uporabo programa v knjižnici na MM_LIBRARY:FIRMANAV.TLB

7 MM_OPER

Na MM_OPER direktoriju je procedura PREPOVED, ki skrbi za prepovedi uporabnikom.

Način klica procedure PREPOVED: `MM_OPER:PREPOVED p1 p2 p3`

p1 - F način, sprememba v UAF pri opisniku
/FLAGS=xxDISUSER

p2 - x oznaka uporabnikov, kot je dovoljena v UAF

p3 - A,P Aktivacija, Prepoved uporabnika

p1 - P način, sprememba v UAF pri posameznih vrstah dela

p2 - x oznaka uporabnikov, kot je dovoljena v UAF

p3 - NBLRDI prepoved vrste dela pri uporabniku, space =
brez prepovedi

n - NETWORK

B - BATCH

l - LOCAL

r - REMOTE

d - DIALUP

i - INTERACTIVE (d+r*1)

Na tem direktoriju je še procedura OBVESTILO, ki napiše sporočilo v pozdravni tekst na ekranu.

Procedure SETPROT in SETUIC nastavljata protekcijo in UIC za potrebe operaterjev.

8 PRIPOROČILA

8.1 Imena datotek

Razlikovali naj bi med dvema skupinama imen:

imena programov in procedur (6 znakov)
imena podatkovnih datotek (6-9) znakov

Vse datoteke naj imajo enako strukturo imena:

anxxxxyyy.ttt ali aanxxxxyyy.ttt

Osnova imena je 6 črk (anxxxx) in je sestavljeno:

- a(a) - eden ali dva znaka aplikacije
lahko je kar prvi znak imena
ali pa je dvoznakovna oznaka aplikacije
- n - način izvajanja programa
programi
 - U - azurirani program
 - P - vnos prometa
 - I - navadna izhodna lista
 - S - sifranti
 vrsta izhodnih list
 - L - letne obdelave
 - M - mesečne obdelave
 - T - tedenske obdelave
 vrsta podatkov
 - D - podatki
 - S - sezname, sifranti
 - B - bazni podatki v sekvenčni obliki
- xxx(x) - oznaka programa ali datoteke v aplikaciji
zaporedna številka (xxx ali xxxx)
govoreča oznaka (xxx ali xxxx)
govoreča oznaka in zaporedna številka (xx-xx ali xxx-xxx)
- ttt - dodatna oznaka za podatkovne datoteke
oznaka poslovne enote

Primer:

programi:	KUKONT	program za azuriranje kontov
	KIKONT	program za izpis kontov
	KSKONT	sifranti kontov

```

ali      GKU001  program za azuriranje kontov
         GKIO01  program za izpis kontov
         GKS001  sifranti kontov

```

imena programov naj bi bila 6 znakov!!

izhodne liste:

```

KUKONT.LIS izhodna lista program KUKONT
KIKONT1.LIS prva izhodna lista KIKONT
KIKONT2.LIS druga izhodna lista KIKONT
KIKONT3.LIS tretja izhodna lista KIKONT

```

Dodani so lahko 3 znaki za dodatni opis datoteke. To naj bi označevalo podatke za pripadnost določeni organizacijski enoti. če je potrebna dodatna delitev, na bi reševali s pomočjo različnih direktorijev (več aplikacij uporablja iste programe). Dogovorjeno mora biti na nivoju sistema.

Tip datoteke opisuje datoteko, kot je predvideno v VMS

```

COB      - cobolski program
EXE      - izvajalni program
LIS      - izhodna lista
COM      - procedura
TMP      - pomožna datoteka
LOG      - dnevnik obdelave
DAT      - sekvenčna podatkovna datoteka
IND      - indeksna podatkovna datoteka
CON,BAS  - bazne datoteke
ttt      - dodatni tipi (dogovorjeni na nivoju sistema)

```

8.2 Logična imena za aplikacijo

Za vsako aplikacijo naj obstajajo logična imena:

```

apli_COB      kobolski programi
apli_EXE      izvajalni programi
apli_COM      procedure
apli_LIB      knjižnice
apli_HLP      pomoč
apli_LOG      dnevniki
apli_DAT      podatki
apli_TMP      pomožni direktorij
apli_LST      izhodne liste

```


3apli_MAS,VAR,VA1 bazni direktoriji

Vsi programi in procedure se sklicujejo samo na logična imena. Aplikacija naj bi bila čim bolj razporejena po vseh diskih na pri aplikaciji moramo tudi standardizirati imena datotek, zapisov in polj ter predpisati obliko polj. Razlikovati moramo med logično in fizično oznako datotek: fizična oznaka je že bila opisana. V COBOLskih programih pa uporabljamo logične oznake datotek, zapisov in polj. Pri tem moramo paziti na omejitve zaradi baze podatkov TOTAL, IDA ali ULTRA, ali katerikoli druge baze. Vsem tem omejitvam zadožča naslednja struktura(?)

ime datoteke	-	4 znaki
ime zapisa	-	ime_datoteke-REC
ime podatka	-	ime_datoteke+4 znaki

primer: ROZD datoteka
ROZD-REC ime zapisa

Aplikacija na bi imela čim več privilegij, instalirano samo na diskih, ki potrebuje izbrane privilegije.
ROZDCTRL imena podatkov
ROZDNAZI
ROZDZIRO....

druga varianta:

Razvoj aplikacije mora imeti enake privilegije kot produkcija.

ime datoteke	-	od 4 do 8 znakov (6)
ime zapisa	-	ime_datoteke-REC
ime podatka	-	ime_datoteke(2-6)+4-6 znakov

Za paketno obdelavo primer: REGOZD datoteka

REGOZD-REC zapis

interaktivni del: 3 RECTRL podatki

znanje priorit: RENAZI paketne obdelave, ki jih

izvedajo interak: REZIRO.....

paketne obdelave: 3

Fizična imena datotek naj bodo enako logičnim.

Aplikacija se mora držati vseh standardov. Vsak odmik mora biti odobren!

Uporabnik naj ima pregled nad obdelavo, reguliranje in kontrola

8.3 Fizični direktoriji za aplikacijo

Obdelava mora biti napisana tako, da se lahko ponavlja brez vračanje zaščitne kopije podatkov.

Vsako logično ime za področje aplikacije naj bo svoj direktorij. Pri tem je potrebno upoštevati: nam datotek se zaščitno kopijo podatkov.

1. velikost

Redno 2. hitrost diska v aplikaciji.

Na sistemskem nivoju je potrebno določiti naslednje stvari, ki se nanašajo na aplikacije in uporabnike:

Delovni čas uporabnikov

Imena uporabnikov in podauporabnikov

Speminjanje in kontroliranje gesel

Delokorg za uporabnike in poduporabnike(menuji)

Sporočanje posebnih stanj sistema in aplikacij izbranim uporabnikom.

Določiti standardne oblike izhodnih list

Določiti paketne vrst in njihove karakteristike za izvajanje paketnih obdelav.

Določiti prioritete in privilegije uporabnikom in aplikacijam.

Prekinitve uporabnikov

8.5 Knjižnice

Vse module za COPY stavke v cobolu lahko damo v tekstovne knjižnice(prihranek prostora, samo ena datoteka)

1. OBJECT knjižnice na nivoju sistema, grupe in procesa. Za povezovanje programov uporabljati 'LNK\$...' logična imena.
2. SHARE knjižnice za vse module, ki se veliko uporabljajo, za tiste, ki se jim spreminja koda ... Ponavadi, ni potrebno ponovno povezovati programov. Knjižnica je stalno instalirana. Pri kreiranju je potrebno paziti na: vrstni red modulov, velikost in kontrolo dostopa. Prihranimo spomin, ker je koda SHARE. S knjižnicami kontroliramo nivo aplikacij.

8.6 Izvajalni programi

Vsi izvajalni programi naj bodo povezani brez DEBUGerja. Izvajalni program je možno instalirati. To je potrebno zaradi privilegijev, ki jih program zahteva, zaradi hitrosti izvajanja, ali zaradi prihranka spomina.

Interaktivne programe je potrebno pisati tako, da se sami končajo, če jih uporabnik dalj časa ne uporablja (15 minut).

Paketne obdelave morajo biti napisane tako, da so čim krajše (10-30 minut). Hitrost je potrebno upoštevati pri oblikovanju aplikacije.

8.7 Ustrezno izbrati produkt za izdelavo aplikacij

1. COBOL
2. generatorji programov
3. generatorji aplikacij
4. baze podatkov
5. CASE orodja

8.8 COBOL

Uporabljati najnovejše verzije COBOLa in ukaze, ki jih omogoča. Posebno pozornost je treba posvetiti konverziji starih programov v nove, ker so se spremenili ANSI standardi (verzija COBOLa 3 -> 4). Aplikacija naj bi imela napisane vse programe v isti verziji Cobola.

8.8.1 COPY stavki v programih -

V programih uporabljati COPY stavke. Vse podatki, ki so stalni, ki so definirani na nivoju sistema, katere bomo spreminjali naj bi bili opisani v COPY modulih, ki jih lahko damo v tekstovno knjižnico, ali pa so samostojne datoteke

8.8.1.1 Opise baze, baznih datotek, klici baze: -

standardna imena za delo z bazo podatkov
 dolzina 5 znakov
 definicija baze podatkov
 imena datotek
 dostopi do baze podatkov
 definicije datotek
 imena polj
 struktura polj(pazi za bodočnost)
 dostopi do baze podatkov
 vrste dostopov

primer:

```
*datoteka kadr_lib:shema.lib-----
01 tfunc          pic xxxxx          value "LOADD".
01 tstat          pic xxxxx          value "      ".
      88 stars          value "****", "BCCR", "LD20".
      88 mrnf          value "MRNF".
      88 bctl          value "BCTL".
      88 held          value "HELD".
      88 nhld          value "NHLD".
      88 held-nhld      value "HELD", "NHLD".
01 tendp          pic xxxxx          value "END.".
01 tenrl          pic xxxxx          value "RLSE".
01 tyend          pic xxxxx          value "RLSE".
01 tadat          pic xxxxx          value "      ".
01 tctrl          pic x(15)          value spaces.
01 tqual          pic xxxxx.
01 shema.
      05 mime          pic x(8)       value "xxxxxxx".
      05 mpro          pic x(6)       value "MYBASE".
      05 mupd          pic x(6)       value "UPDATE".
      05              pic xx         value "NL".
      05              pic x(12)      value "ORENSHRE".
      05              pic x(12)      value "RAZPSHRE".
      05              pic x(12)      value "KADRSHRE".
      05              pic x(12)      value "xxxxxxxxx".
      05              pic x(4)       value "END.".
*datoteka kadr_lib:kadr.lib-----
```

```

01 kadrt.
    10 pic x(32) value "KADRCTRLKADRNAZIKADRNASL".
    10 pic x(8)  value "END.".
01 kadra.
    10 kadrctrl      pic x(5).
    10 kadrnazi      pic x(30).
    10 kadrnasl      pic x(30).
    10 kadrosta      pic x(xx).
*datoteka kadr_lib:razp.lib-----
01 razpkak          pic x(5). /kljuc za link KADRLKZP
01 razpork          pic x(5). /kljuc za link ORENLKZP
01 razpkar          pic x(4). /reference za link KADRLKZP
01 razporr          pic x(4). /reference za link ORENLKZP
01 razpt.
    10 pic x(24) value "RAZPKADRRAZPORENRAZPOSTA".
    10 pic x(8)  value "END.".
01 razpa.
    10 razpkadr      pic x(5).
    10 razporen      pic x(5).
    10 razposta      pic x(x).
*datoteka kadr_lib:qkadr.lib-----
*klica za dototeko KADR (vse operacije in RDNXT)
qkadr. call "mmbase" using tfunc tstat "KADR" kadrctrl
        kadrt      kadra tyend.
        move tenr1 to tyend move kadrctrl to tctrl
        move "kadr" to tadat.
qkadr. call "mmbase" using tfunc tstat "KADR"
        tqual kadrt kadra tyend.
        move tenr1 to tyend move kadra to tctrl
        move "KADR" to tadat.
*datoteka kadr_lib:qrazp.lib
*klici za daoteko RAZP ( vse oprracije in RDXNT)
qrazpor.call "mmbase" using tfunc tstat "RAZP"
        razporr "orenlkzp" razpork razpt razpa tyend.
        move tenr1 to tyend move razpork to tctrl
        move "RAZP" to tadat.
qrazpka.call "mmbase" using tfunc tstat "RAZP"
        razpkar "kadrlkzp" razpkak razpt razpa tyend.
        move tenr1 to tyend move razpkak to tctrl
        move "razp" to tadat.
qrazpr. call "MMBASE" using tfunc tstat "RAZP"
        tqual razpt razpa tyend.
        move tenr1 to tyend move RAZPa to tctrl
        move "RAZP" to tadat.
-----

```

Pri bazi je potrebno razdelati razne zanke pri branju ali pri ažuriranju:

1. zadrževanje zapisov
2. zakljenjeni zapisi

Primer zadržanega in zakljenjenega zapisa:

```

move "HELD" to tstat
perform until not held-nhld
    move "funcija" to to tfunc
    move zapis to bazni-zapis
    move "LINK" to link-polje
    perform klic-baze
end-perform

```

8.8.1.2 Opise RMS datotek -

Opis SELECT stavka za datoteke:

1. naslov datoteke
2. dostopi
3. opis datoteke
4. klici datoteke

```

primer datoteke kdarovske evidence
*SELECT DATOTEKE "kadr_lib:skadr.lib"-----
*select KADR assign to "kadr_bas:kadr.ind"
    organization is indexed
    record key is kdrctrl
    status kadr-stts access mode is dynamic.
*opis datoteke KADR- "kadr_lib:kadr.lib"-----
*opis za FD
FD kadr recors varying depeneding on kadr-dolz.
01 kadra.
    10 kdrctrl      pic x(5).
    10 kadrnazi     pic x(30).
    10 kadrnasl     pic x(30).
    10 kadrosta     pic x(xx).
*opis za workins storage section "kadr_lib:wkadr.lib"
01 kadr-stts      pic xx.
01 kadr-dolz      pic 9(4) comp.
*DECLARATIVES   "kadr_lib:dkadr.lib"-----
kadr-decl section.
    use after error procedure on kadr.
    *testiranje statusa ob raznih akcijah
*-----

```

8.8.1.3 Ekranski del -

1. splošni podatki za ekran
2. krmiljenje ekrana
3. klici ekrana

```

*datoteka "kadr_lib:ekran.lib"-----
01 parametri.
   05 izpis    pic x(40).
   05 tekst    pic x(80).
   05 tvpis.
       10 tznak      pic x.
           88 vredit value "&".
           88 nazaj   value "<".
           88 ven     value "#".
           88 da      value "d" "D".
           88 ne      value "n" "N".
   05 ierr     pic 999.
   05 scroll    pic x.
   05 crtice    pic xx.
       88 vnos    value "d".
   05 wdatm     pic x(6).
   05 wnumr     pic s9(15) comp-3.
   05 wndcp     pic 99.
   05 kontrola.
       10 filler    pic x(5).
       10 tipka     PIC 9.
       10 filler    pic x(6).
*datoteka za krmiljenje "kadr_lib:program1.lib"-
01 WTDIS.
   05 W01 PIC X(40) VALUE "2401=NE MOREŠ NAPREJ - KLIČI ERC!' >01 ".
   05 W02 PIC X(40) VALUE "0101#...SREČNO!...' 002".
   05 W03 PIC X(40) VALUE "0101#*LB-GBL*' +004".
   05 W04 PIC X(40) VALUE "0225#AZURIRANJE VIRTERJA' 005".
   05 W05 pic x(40) value "0501 1.Virtualni naslov : VT' >03I".
   05 W06 pic x(40) value "0601 2.Fizična oznaka vrat : ' >08".
   05 W07 pic x(40) value "0801 4.Tip terminala : ' >01".
   05 W08 PIC X(40) VALUE "0901 5.Sifra poslovne enote : ' >01".
*klici "kadr_lib:qekran.lib"-----
qwrts section 1.
qwrts.
   if ierr = zero move wadis(indx) to izpis
   else          move wadis(ierr) to izpis.
   call "xekran" using parametri.
   if t = 2 go to qwrts.
   if t = 9 stop run.
qwrte. exit.
*-----

```

8.8.1.4 Standardni deli programa -

1. definicije podatkov
2. razna ponavljajoča koda

```
*wokring "kadr_lib:wss.lib"-----
01 razni-stevci.
    10 x          pic s9(9) comp.
    10 y          pic s9(9) comp.
    10 w          pic s9(9) comp.
    10 wrtrn     pic s9(9) comp.
*zacetek "kadr_lib:zacetek.lib"-----
zacetek.
    move 0 to t ierr.
    move "d" to crtice scroll.
    go to a00.
konec.  move 2 to indx.
        move 9 to t.
        perform qwrt.
        stop run.
*-----
```

8.8.2 Sortiranje z SORT INPUT in OUTPUT PROCEDURE -

Podprogram za sortiranje, katerega uporaba ne podle strukture programa.

```
*definicija podatkov
*datoteka "kadr_lib:sort.lib"-----
01 xxsorststev  pic 9(4) comp value 1.
01 xxsortfunc   pic x.
01 xxsortdata.
    10 xxsortsize  pic 9(4) comp.
    10 xxsortkeys occurs 9.
    20 xxsortascd  pic 9(4) comp.
        20 xxsortstar  pic 9(4) comp.
        20 xxsordolz  pic 9(4) comp.
    10            pic x(482).
01 xxsortdatar redefines xxsortdata.
    10 data1      pic x.
    10 data6      pic x(6).
    10 datax.
        20 dctrl      pic x(5).
        20 dprim      pic x(20).
        20 ddroj      pic x(11).
```



```

                20 dsmer          pic x(20).
                20                pic x.
*nastavitev sorta-----
call "xxsort" using xxsortstev "F" xxsortdata giving wrtrn
*input za sortiranje-----
call "xxsort" using xxsortstev "I" xxsortdata giving wrtrn
*sortiranje-----
call "xxsort" using xxsortstev "S" xxsortdata giving wrtrn
*output iz sortiranja-----
call "xxsort" using xxsortstev "O" xxsortdata giving wrtrn
*-----

```

8.8.3 Za sortiranje tabel -

Podprogram za sortiranje tabel v programu:

```

*parametri "kadr_lib:xxsota.lib"-----
01 xxsota-par.
    10 stevilo-elementov      pic 9(4) comp.
    10 dolzina-elementa      pic 9(4) comp.
    10 pozicija-sort         pic 9(4) comp.
    10 dolzina-sort          pic 9(4) comp.
01 tabela.
    10 element                pic x(60) occurs 100.
*sortiranje
    call "xxsota" using xxsota-par tabela.

```

8.8.4 podatki -

Določiti namen datoteke (interaktivno, batch aziriranje)

1. dostop do datoteke (read, write, rewrite)
2. strukturo zapisov (fiskni, variabilni, verige, dostop po različnih ključih)
3. hitrost dostopa (do enega zapisa, do več zapisov..)
4. dostop večih uporabnikov do datoteke
5. prenačanje datoteke (velikost, iz računalnika na računalnik (IBM!))
6. restavriranje (obnavljanje) datoteke

Doseči je potrebno:

1. najvišjo hitrost dostopa
2. najmanjšo porabo prostora, spomina, procesorja
3. enostavnost strukture
4. enostavno obnavljanje datoteke
5. varnost ačuriranja
6. prenosljivost
7. možnost širjenja zapisov, širjenje datoteke

predlogi:

1. uporabljati RMS datoteke za vse podatke, ki se ne spreminajo, ali pa jih spreminjajo BATCH obdelave
2. določiti vrste dostopov do datoteke (samo read, ali samo write ...)
3. določiti število dostopov do datoteke (čim manj uporabnikov na enkrat)
4. uporabljati variabilno dolžino zapisov (hitrost)
5. določiti več dostopnih poti za dostop do zapisa (hitrost)
6. redno obnavljanje
7. enostavno dodajanje novih podatkov v zapis

8.9 Prednosti in slabosti:

8.9.1 RMS datoteke: -

1. Software je v operacijskem sistemu
2. enostavne
3. neomejena dolžina(!?)
4. hitro sekvenčno branje
5. možnost dostopa po večih (delnih) kjučih
6. enostavno obnavljanje
7. dolg dostop do izbranega zapisa
8. težko ročno zaklepanje in kontrola zaklepanja
9. težko vrivanje zapisov

10. ni sortiranja po delnih ključih v okviru ključa

8.9.2 Baze podatkov: -

1. software je potrebno kupiti
2. transakcijsko logiranje
3. omejena dolžina datoteke
4. hitro branje po ključu
5. veriženje zapisov
6. enostavno zaklepanje
7. zelo malo delnih ključev(alii nič)
8. težko in počasnejše obnavljanje(!?)
9. relacijske ali mrežne
10. poraba spomina in procesorja

8.10 Izhodne liste

1. izhodne liste naj bi imele tri širine: 80,110,132 znakov
standardno dolžino strani: 24,36,64,72 vrstic
2. standardna glava
3. možnost prenosa na FICHE
4. za preskok na novo stran uporabljati <FF>
5. za prazne vrstice uporabljati preskok vrstic
6. variabilna dolžina zapisov

8.11 Ekranse slike(interaktivno delo)

1. uporabljati enak način obdelave (glave, menuji, potek programa)
2. uporabljati standardne tipke (pomoč,potrditev, prekinitev, prejšnje polje, naslednje polje,kalkulator....)
3. uporabljati čim manj tipk

4. definiranje ekranske slike zunaj programa
5. enostavno popravljanje slike
6. enostaven klic slike
7. možnost ekranskega prerisa na tiskalnik ali datoteko
8. prilagoditi med seboj: obliko ekrana, promet po liniji, prekinitve procesorja

N A V O D I L A Z A I Z D E L A V O I N S P R E M I N J A N J E M E N U J E V Z M M - o k o l j e m

MM-okolje se povezuje z operacijskim sistemom na nivoju userjev. To pomeni, da takoj, ko se prijavimo na sistem MM-okolje preveri, če kot uporabnik obstojamo. Naprej pa smo potem vodeni z okoljem. Kot user - aplikacija imamo lahko več uporabnikov. Eni vpisujejo, drugi ažurirajo, tretji pregledujejo oz. listajo en del podatkov, četrti pa neki drug del itd... Vse to uravnavamo z menuji, ki so vezani na določenega uporabnika. Nivojev menujev je lahko do 8. Pri kreiranju nove aplikacije je potrebno najprej ažurirati komandni niz MMPARAM.COM na direktoriju MM_PARAM, kjer je definirana tabela logičnih imen, in dodati ime aplikacije v datoteko MM_PARAM:APLIK.PAR. Na koncu še startamo aplikacijo z ZMM_MANAGER:STARTAPLI 'APLI'. To ažurira sistemski programer.

Spreminjanje menujev pa je delo organizatorja oz. skrbnika aplikacije.

Postopek teče takole:

Na sistem se prijavimo z X'APLIK'.

Poženemo program MM_SYSTEM:AZUMEN, kjer moramo vpisati:

1. nivo - USER:	username	(n.pr.: KUPCI)
	indikator	- če je 01 uporabnik lahko dela sicer ne
	O.e.(org.enota)	
	naziv userja	- tekst
	ime vozla	- prazno
	diski	- prazno
	aplikacija	- ime aplikacije v seznamu aplikacij (kot je v MM_PARAM.COM - 4 znake)
	baza	- prazno
	stop baze	- prazno
	datpar1	- prazno
	seznam	- prazno
	ind.byti	- prazno

2. nivo - UPORABNIK: uporabnik - ime uporabnika

indikator - če je 01 uporabnik lahko dela
 0.e.(org.enota)

naziv upor. - tekst

telefon - int.stev. telefona

geslo

menu - ime menuja - obvezno
 (če imamo PRG si ime menuja zmislimo)

def.dir. - prazno

datpar1 - prazno

seznam - prazno

ind. byti - E če imamo PRG sicer prazno

3. nivo - MENU:

program - ime programa

oznaka - oznaka v menuju za uporabnika

opis - tekst

način - način izvajanja n.pr.:
 Zskli - če je komandna procedura,
 PRG:skli - če je prg in
 r skli - če je izvajalni program

prioriteta - če pustimo prazno ostane default = 4

indikator - če je 01 se program oz. proc.
 lahko izvaja sicer ne

Če dodajamo novega uporabnika (terminal) na sistem moramo pognati še program MM_SYSTEM:VIRTER, kjer določimo kateri terminali lahko izvajajo posamezne aplikacije.

Obstoja že nov program MMOKOL, ki združuje oba zgornja, vendar ima še precej napak, ki jih je potrebno odpraviti. Datoteke s katerimi delajo programi so iste, zato je vseeno če sedaj uporabljamo oba zgornja programa, ko bo nov popravljen, pa bomo uporabljali novega.