EUMEL

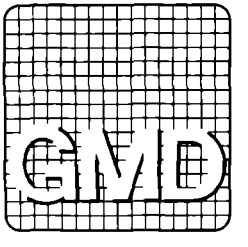
Extendable multi User Microprocessor ELAN-system

Version 1.6

TASCHENBUCH

Hochschul-
Rechen-
Zentrum

Universität Bielefeld



Inhaltsverzeichnis

Kommunikation mit dem Supervisor

Kontrolltasten: (Realisierung Terminal-abhängig)

<STOP>	hält die Terminal-Ausgabe an
<WEITER>	läßt die Terminal-Ausgabe weiterlaufen
<SV>	Anruf an den Supervisor Multi-User: Falls das Terminal nicht schon im Supervisor-Modus ist, schaltet das System dahin um, so daß Supervisor-Kommandos gegeben werden können. Single-User: Abbruch des laufenden Benutzer-Programms (wirkt wie <SV> 'halt' RETURN im Multi-User)

Supervisor-Kommandos:

```
begin ("TASK") ............. neue Task als Sohn von "PUBLIC" anfangen
begin ("TASK","FATHER") .... neue Task als Sohn von "FATHER" anfangen
end ........................ aktuelle Task beenden (Dateien löschen)

break ...................... aktuelle Task vom Terminal abkoppeln
continue ("TASK") .......... "TASK" an das Terminal ankoppeln

continue ................... vom SV-Modus zur noch angekoppelten (!)
                             Task zurückschalten
halt ....................... laufendes Programm abbrechen
```

Monitor-Kommandos

Kommandos im Multi- und Single-User-System:

Informations-Kommandos

```
help ..................... Anzeige der am häufigsten benutzten Kommandos
storage info ............. Speicherplatz des externen Speichers
```

Editor-Kommandos (vergl. Kapitel 4):

```
edit ("datei") ........... Datei edieren
edit ("datei1","datei2") ... Parallel-Editor
editmode ................. Fließtext ein-/ausschalten
```

Compiler-Kommandos (vergl. Kapitel 5):

```
run ("datei") ............ Übersetzen und ausführen eines ELAN-Programms
run again ................ Letztes übersetztes Programm nochmal ausführen
insert ................... Übersetztes Programm eintragen
```

Datei-Kommandos (vergl. Kapitel 6 und 7):

```
copy ("datei", "duplikat") . Datei kopieren
rename ("alt", "neu") ..... Datei umbenennen
reorganize ("datei") ..... Datei "reorganisieren"
forget ("datei") ......... Datei löschen
killer ................... Mehrere Dateien löschen
list ..................... Dateien anzeigen
```

Archiv-Kommandos (vergl. Kapitel 8):

```
archive ("name") ......... Archiv anmelden
clear archive ............ Löscht ein Archiv
save archive ............. Archiviert alle Dateien einer Task
load archive ............. Holt alle Dateien eines Archivs in eine Task
to archive ("datei") ..... Datei archivieren
from archive ("datei") ..... Datei vom Archiv holen
list archive ............. Dateien des Archivs anzeigen
```

Textkosmetik und Drucker (vergl. Kapitel 9 und 11):

```
lineform ("datei") ....... Zeilenweises Formatieren
pageform ("datei", "raus") . Seitenweises Formatieren
print ("datei") ........... Drucken
```

Nur im Single-User-System:

shutup ..... Ende einer Sitzung

Nur im Multi-User-System:

end..... Task und Dateien löschen
break..... Task abkoppeln
task info ..... Information über alle aktiven Tasks

list ("MANAGER") ..... Dateien eines Managers anzeigen

Editor-Kommandos

Tastenfunktionen:

→, ←, ↓, ↑	zeichenweises Positionieren
HOP→, HOP←, HOP↓, HOP↑	zeilen- und bildweises Positionieren
RETURN	zum nächsten Zeilenanfang, evtl. einrücken
HOP RETURN	nächstes Bild, aktuelle Zeile wird 1.Bildzeile
RUBOUT	Zeichen löschen
HOP RUBOUT	X : Zeilenrest löschen ; X : Zeile löschen ; X : rückbrechen
RUBIN	RUBIN Zustand ein/aus
HOP RUBIN	INS Zustand (Zeilen einfügen) ein/aus; aufbrechen, falls Cursor innerhalb der Zeile
TAB	zur nächsten Tabulatorposition; zum logischen Zeilenanfang /-ende
HOP TAB	Tabulator setzen/löschen
MARK	Markierung ein/aus
ESC ESC	Kommandodialog
ESC <Taste>	<Taste> ausführen

Lernen:

HOP ESC <zu lernende Sequenz> HOP ESC <Taste>

Vorbesetzte Tasten:

ESC q	Editor verlassen.
ESC w	Wechseln (* nur beim Parallel-Editor *).
ESC f	Letztes Kommando wiederholen
ESC k	Letztes Kommando wieder darstellen
ESC 1	Zum Dateianfang
ESC 9	Zum Dateiende
ESC v	Auf volle Bildgröße schalten (* auch beim Parallel-Editor *)
ESC h	Auf halbe Bildgröße schalten
ESC d	duplizieren: IF markiert THEN schreibe markiertes in scratch datei ELSE hole scratch datei FI ; demarkiere .
ESC p	schreibe markiertes in scratch datei ; lösche markiertes .
ESC g	hole scratch datei ; demarkiere .
ESC z	zentriere aktuelle zeile .
ESC A	Ä
ESC a	ä
ESC O	Ö
ESC o	ö
ESC U	Ü
ESC u	ü
ESC B	ß

Dateieditor-Kommandos:

BEGIN <zahl>	linke Bildgrenze setzen (Standard: 1)
LIMIT <zahl>	rechte Zeilegrenze (für Umbruch) setzen (Standard: 77)
quit	Editor verlassen
<zahl>	auf Zeile <zahl> positionieren
+ <zahl>	um <zahl> Zeilen vorwaerts positionieren
- <zahl>	um <zahl> Zeilen rückwärts positionieren
"<muster>"	vorwärts nach <muster> suchen
"<alt>" CHANGETO "<neu>" "<alt>" C "<neu>"	vorwärts nach <alt> suchen, durch <neu> ersetzen
GET "<datei>"	Inhalt von <datei> einfügen, das Eingefügte bleibt markiert
PUT "<datei>"	Markiertes in <datei> schreiben ("" ist Scratchdatei)
REP....ENDREP	Schleife, 'eof' als Endekriterium

Weitere Dateieditor-Kommandos für den Programmierer:

MARK 1	Markierung einschalten
MARK 0	Markierung ausschalten
HALT <zahl>	nächste Suche nur bis Zeile <zahl>
WRITE "<Text>"	mit <Text> Editor-Eingabe simulieren
COL <zahl>	auf Spalte <zahl> positionieren
SIZE <zahl>	Bildgröße auf <zahl> Zeilen einstellen; Standard: 23
col	liefert aktuelle Spaltenposition
row	liefert aktuelle Zeilenposition
len	liefert Länge der aktuellen Zeile
limit	liefert aktuell eingestellte Zeilengrenze
size	liefert aktuelle Bildgröße
mark	liefert 1 falls Markierzustand, sonst 0
:= , + , -	übliche INT bzw. TEXT Operationen
REP....UNTIL..ENDREP	Schleife mit explizitem Endekriterium
IF..THEN....FI	einseitige Bedingung
IF..THEN....ELSE....FI	zweiseitige Bedingung

Variablen können ohne Deklaration verwandt werden (maximal 10 Stück), Klammern können in Ausdrücken wie gewohnt benutzt werden. Anstelle von <zahl> kann immer ein INT-Ausdruck, anstelle von <text> ein TEXT-Ausdruck verwandt werden. Alle Operatoren können abgekürzt werden.

Textbe- und -verarbeitung

Kommandos für 'lineform':

```
bsp ..................... Zeichen übereinander drucken
command ("ß", "ß") ..... Kommandozeichen einstellen (z.B. 'ß')
limit (72) .............. Anzahl Zeichen/Zeile (z.B. 72)
type ("pica") ........... Schrifttyp einstellen (z.B. auf 'pica')
```

Kommandos für 'pageform':

```
bottom .................. Fußzeilen u. Seitennummern
bottomeven ..............          "
bottomodd ...............          "
bottomend ...............          "
command ("ß", "ß") ..... Kommandozeichen einstellen (z.B. 'ß')
foot .................... Fußnote
footend .................    "
free (1.5) .............. Zeilen freilassen (z.B. ein und eine halbe)
head .................... Kopfzeilen u. Seitennummern
headeven ................          "
headodd .................          "
headend .................          "
linefeed (2.0)........... Zeilenabstand (z.B. zweizeilig)
page .................... Neue Seite anfangen
pagelength (40) ......... Zeilen/Seite setzen (z.B. 40 Zeilen/Seite)
pagenr ("%", 3)..........Seitennr bzw. -zeichen
```

Kommandos für den EUMEL-Drucker:

```
block ................... Randausgleich
bsp ..................... Zeichen übereinander drucken
command ("ß", "ß") ..... Kommandozeichen einstellen (z.B. 'ß')
free (3.0)............... Zeilen freilassen (z.B. 3 Zeilen)
limit (72)............... Anzahl Zeichen/Zeile (z.B. 72)
linefeed (2.0)........... Zeilenabstand (z.B. zweizeiligen)
page .................... Neue Seite anfangen
pagelength (50).......... Schreibfeld (Zeilen/Seite) einstellen
start (3, 4)............. Schreibfeld positionieren (z.B. drei Blanks
                          und vier Zeilen von der oberen linken Ecke)
type ("elite") .......... Schrifttyp einstellen (z.B. 'elite')
ub ...................... Unterstreichung einschalten
ue ...................... Unterstreichung ausschalten
```

EUMEL-Zeichencode

Steuerzeichen und -tasten:

Der EUMEL-Standard definiert einen Satz von (Ausgabe-)Steuerzeichen mit ihren (ausgabeseitigen) Wirkungen. Diese Steuerzeichen sind geräteunabhängig und werden vom EUMEL-System automatisch für die jeweiligen Geräte passend umcodiert. Bei Standard-Geräteschnittstellen ist die Wirkung anderer Steuerzeichen nicht definiert. Für den Anschluß von Drucker, für Datenfernübertragung u.ä. können sogenannte "Transparent"-Schnittstellen verwandt werden. Bei diesen Schnittstellen ist die Wirkung aller Steuerzeichen undefiniert (vom angeschlossenen Gerät abhängig); es wird aber garantiert, daß alle Zeichen ohne Code-Umsetzung direkt ausgegeben werden.

Standardsatz der Ausgabesteuerzeichen:

Wert	Bezeichnung	Wirkung
0	NUL	keine Wirkung, Füllzeichen
1	HOME	Cursor auf linke obere Ecke des Bildschirms
2	→	Cursor eine Stelle nach rechts
3	↑	Cursor eine Zeile nach oben
4	CL EOP	Löschen von Cursor-Position bis Bildschirmende
5	CL EOL	Löschen von Cursor-Position bis Zeilenende
6	CPOS	Cursor positionieren, nächstes Ausgabezeichen bestimmt die y-Position (0 <= code(y) <= 23), darauffolgendes Ausgabezeichen die x-Position (0 <= code (x) <= 78).
7	BELL	akustisches Signal
8	←	Cursor eine Stelle nach links
10	↓	Cursor eine Zeile nach unten, bzw. 'roll up' falls der Cursor schon in der letzten Zeile stand
13	RETURN	Cursor an den Anfang der aktuellen Zeile
14	ENDMARK	Ende des zu markierenden Bereichs
15	BEGINMARK	Anfang des zu markierenden Bereichs

Für die Eingabeseite ist im EUMEL ein Standardsatz von Steuertasten definiert. Diese Tasten sollten am angeschlossenen Gerät vorhanden sein oder bereitgestellt werden (z.B. durch Überkleben von Tasten) oder müssen mit Hilfe der CONTROL-Taste simuliert werden. Das EUMEL-System führt entsprechend der folgenden Tabelle evtl. notwendige Umcodierungen der Eingabe durch. Evtl weitere vorhandenen Spezialtasten erzeugen gerätespezifische Codes. Bei der "Transparent"-Schnittstelle werden - symmetrisch zur Ausgabeseite - alle hereinkommenden Zeichen ohne Codeumsetzung weitergereicht.

Standardsatz der Eingabe-Steuertasten:

Wert	Bezeichnung
1	HOP
2	→
3	↑
8	←
9	TAB
10	↓
11	RUBIN
12	RUBOUT
13	RETURN
16	MARK
27	ESC

Darstellbare Zeichen

Die genaue Darstellung der einzelnen Zeichen hängt vom Endgerät ab. Die Zeichen mit den Codes 32 - 125 sind i. A. auf allen Geräten vorhanden, ebenso der Trennstrich (173). Andere Zeichen sind evtl. nur auf Spezialgeräten verfügbar.

Beispiele zum Lesen der Tabelle:

```
code (" ") -> 32
code ("m") -> 109
```

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0		(	2	<	F	P	Z	d	n	x													
1		)	3	=	G	Q	[	e	o	y													
2	␣	*	4	>	H	R	!	f	p	z													
3	!	+	5	?	I	S	]	g	q	{					-		Ä		Ü				
4	"	,	6	@	J	T	^	h	r	!							ß						
5	#	-	7	A	K	U	_	i	s	}										ä		ü	
6	$	.	8	B	L	V	`	j	t	~													
7	%	/	9	C	M	W	a	k	u									ö					
8	&	0	:	D	N	X	b	l	v														
9	'	1	;	E	O	Y	c	m	w												ß		

Bulletin (nach Paketen):

In dieser Liste sind alle verfügbaren Prozeduren, Operatoren und Datentypen des EUMEL-Systems aufgeführt, einschließlich der Prozeduren von System-Programmen und Prozeduren, die verübergehend bzw. versuchsweise in das System übernommen wurden. Die letzteren Prozeduren sind -außer an dieser Stelle - nicht näher beschrieben und können ohne weiteres jederzeit geändert werden. Benutzung erfolgt also auf eigene Gefahr.

Die mit
    (M) sind nur im Multi-User, während die mit
    (+) gekennzeichneten Objekte nicht standardmäßig vorübersetzt sind.
Dabei bedeuten bei den formalen Parametern
    C CONST und
    V VAR.

PACKET text :

```
TEXT ......... TYPE
:= ........... (TEXT V dest, TEXT C source)
= ............ (TEXT C left, right) --> BOOL
<> ........... (TEXT C left, right) --> BOOL
< ............ (TEXT C left, right) --> BOOL
<= ........... (TEXT C left, right) --> BOOL
> ............ (TEXT C left, right) --> BOOL
>= ........... (TEXT C left, right) --> BOOL
maxtextlength . --> INT
SUB ........... (TEXT C text, INT C pos) --> TEXT
subtext ....... (TEXT C source, INT C from, to) --> TEXT
subtext ....... (TEXT C source, INT C from) --> TEXT
code .......... (TEXT C text) --> INT
code .......... (INT C code) --> TEXT
ISUB .......... (TEXT C text, INT C index) --> INT
replace ....... (TEXT V text, INT C index, value)
RSUB .......... (TEXT C text, INT C index) --> REAL
replace ....... (TEXT V text, INT C index, REAL C code)
replace ....... (TEXT V dest, INT C pos, TEXT C source)
text .......... (TEXT C source, INT C length) --> TEXT
text .......... (TEXT C source, INT C length, from) --> TEXT
CAT ........... (TEXT V right, TEXT C left)
+ ............. (TEXT C left, right) --> TEXT
* ............. (INT C times, TEXT C source) --> TEXT
length ........ (TEXT C text) --> INT
LENGTH ........ (TEXT C text) --> INT
pos ........... (TEXT C source, pattern) --> INT
pos ........... (TEXT C source, pattern, INT C from) --> INT
pos ........... (TEXT C source, pattern, INT C from, to) --> INT
pos ........... (TEXT C source, low, high, INT C from) --> INT
compress ...... (TEXT C text) --> TEXT
```

```
change ........ (TEXT V destination, INT C from, to, TEXT C new)
change ........ (TEXT V destination, TEXT C old, new)
changeall ..... (TEXT V destination, TEXT C old, new)
deletechar .... (TEXT V string, INT C deletepos)
insertchar .... (TEXT V string, TEXT C char, INT C insertpos)
heapsize ...... --> INT
```

```
PACKET   basictransput  :

out ........... (TEXT C text)
out ........... (TEXT C eins, zwei)
out ........... (TEXT C eins, zwei, drei)
out ........... (TEXT C eins, zwei, drei, vier)
outsubtext .... (TEXT C source, INT C from)
outsubtext .... (TEXT C source, INT C from, to)
TIMESOUT ...... (INT C times, TEXT C text)
inchar ........ (TEXT V character)
incharety ..... --> TEXT
incharety ..... (INT C timelimit) --> TEXT
pause ......... (INT C timelimit)
put ........... (TEXT C text)
putline ....... (TEXT C text)
line ..........
line .......... (INT C number)
page ..........
cursor ........ (INT C x, y)
getcursor ..... (INT V x, y)
cout .......... (INT C number)
```

```
PACKET   bool  :

BOOL .......... TYPE
:= ............ (BOOL V dest, BOOL C source)
NOT ........... (BOOL C operand) --> BOOL
AND ........... (BOOL C left, right) --> BOOL
OR ............ (BOOL C left, right) --> BOOL
CAND .......... (BOOL C left, right) --> BOOL
COR ........... (BOOL C left, right) --> BOOL
XOR ........... (BOOL C left, right) --> BOOL
```

```
PACKET   integer  :


INT ........... TYPE
:= ............ (INT V dest, INT C source)
maxint ........ --> INT
+ ............. (INT C left, right) --> INT
- ............. (INT C left, right) --> INT
* ............. (INT C left, right) --> INT
DIV ........... (INT C left, right) --> INT
MOD ........... (INT C left, right) --> INT
- ............. (INT C operand) --> INT
INCR .......... (INT V dest, INT C increment)
DECR .......... (INT V dest, INT C decrement)
= ............. (INT C left, right) --> BOOL
<> ............ (INT C left, right) --> BOOL
< ............. (INT C left, right) --> BOOL
<= ............ (INT C left, right) --> BOOL
> ............. (INT C left, right) --> BOOL
>= ............ (INT C left, right) --> BOOL
sign .......... (INT C argument) --> INT
SIGN .......... (INT C argument) --> INT
abs ........... (INT C argument) --> INT
ABS ........... (INT C argument) --> INT
** ............ (INT C arg, exp) --> INT
min ........... (INT C first, second) --> INT
max ........... (INT C first, second) --> INT
text .......... (INT C number) --> TEXT
put ........... (INT C number)
text .......... (INT C number, length) --> TEXT
int ........... (TEXT C number) --> INT
lastconversionok --> BOOL
setconversion . (BOOL C success)
get ........... (INT V number)
random ........ (INT C ugrenze, ogrenze) --> INT
initializerandom (INT C wert)


possible errors :INT OP ** : negative exponent
                 0 ** 0 is not defined
-------------------------------------------------------------------------




PACKET   pcbandinitconnrol  :


INITFLAG ...... TYPE
pcb ........... (INT C field) --> INT
setlinenr ..... (INT C value)
initialized ... (INITFLAG V flag) --> BOOL
clock ......... (INT C nr) --> REAL
-------------------------------------------------------------------------
```

```
PACKET   errorhandling  :


enablestop ....
disablestop ...
iserror ....... --> BOOL
clearerror ....
errormessage .. --> TEXT
errorline ..... --> INT
errorstop ..... (TEXT C message)
puterror ......
stop ..........


possible errors :stop
-----------------------------------------------------------------------


PACKET   feldeditor  :


editget ....... (TEXT V editsatz, INT C editlimit, editfende)
editget ....... (TEXT V editsatz)
-----------------------------------------------------------------------


PACKET   commanddialogue  :


commanddialogue  --> BOOL
commanddialogue  (BOOL C status)
yes ........... (TEXT C question) --> BOOL
no ............ (TEXT C question) --> BOOL
say ........... (TEXT C message)
paramposition . (INT C x)
lastparam ..... --> TEXT
lastparam ..... (TEXT C new)
-----------------------------------------------------------------------


PACKET   input  :


get ........... (TEXT V word)
get ........... (TEXT V word, TEXT C separator)
get ........... (TEXT V word, INT C length)
getline ....... (TEXT V line)
getsecretline . (TEXT V line)
-----------------------------------------------------------------------
```

```
PACKET   real  :


REAL ..... TYPE
maxreal ..... --> REAL
smallreal ..... --> REAL
:= ..... (REAL V dest, REAL C source)
+ ..... (REAL C left, right) --> REAL
- ..... (REAL C left, right) --> REAL
* ..... (REAL C left, right) --> REAL
/ ..... (REAL C left, right) --> REAL
- ..... (REAL C operand) --> REAL
= ..... (REAL C left, right) --> BOOL
<> ..... (REAL C left, right) --> BOOL
< ..... (REAL C left, right) --> BOOL
<= ..... (REAL C left, right) --> BOOL
> ..... (REAL C left, right) --> BOOL
>= ..... (REAL C left, right) --> BOOL
floor ..... (REAL C real) --> REAL
round ..... (REAL C real, INT C digits) --> REAL
text ..... (REAL C real) --> TEXT
put ..... (REAL C real)
text ..... (REAL C real, INT C length, fracs) --> TEXT
real ..... (TEXT C text) --> REAL
get ..... (REAL V value)
abs ..... (REAL C value) --> REAL
ABS ..... (REAL C value) --> REAL
sign ..... (REAL C value) --> INT
SIGN ..... (REAL C value) --> INT
MOD ..... (REAL C left, right) --> REAL
frac ..... (REAL C value) --> REAL
max ..... (REAL C a, b) --> REAL
min ..... (REAL C a, b) --> REAL
INCR ..... (REAL V dest, REAL C increment)
DECR ..... (REAL V dest, REAL C decrement)
int ..... (REAL C value) --> INT
real ..... (INT C value) --> REAL
------------------------------------------------------------------------
```

```
PACKET   datehandling  :


date ..........  --> TEXT
date ..........  (REAL C datum) --> TEXT
timeofday .....  --> TEXT
timeofday .....  (REAL C value) --> TEXT
time ..........  (REAL C value) --> TEXT
date ..........  (TEXT C datum) --> REAL
time ..........  (TEXT C time) --> REAL


possible errors :negativ date is not allowed !
-----------------------------------------------------------------------


PACKET   dataspace  :


:= .............  (DATASPACE V dest, DATASPACE C source)
nilspace .......  --> DATASPACE
forget .........  (DATASPACE V dataspace)
type ...........  (DATASPACE C ds, INT C type)
type ...........  (DATASPACE C ds) --> INT
-----------------------------------------------------------------------


PACKET   thesaurushandling  :


THESAURUS .....  TYPE
emptythesaurus   --> INITIALTHESAURUS
:= .............  (THESAURUS V dest, INITIALTHESAURUS C dummy)
insert .........  (THESAURUS V thesaurus, TEXT C name, INT V index)
delete .........  (THESAURUS V thesaurus, TEXT C name, INT V index)
delete .........  (THESAURUS V thesaurus, INT C index)
CONTAINS .......  (THESAURUS V thesaurus, TEXT C name) --> BOOL
rename .........  (THESAURUS V thesaurus, TEXT C old, new)
isfull .........  (THESAURUS C thesaurus) --> BOOL
link ...........  (THESAURUS V thesaurus, TEXT C name) --> INT
name ...........  (THESAURUS C thesaurus, INT C index) --> TEXT
highestentry ...  (THESAURUS C thesaurus) --> INT
-----------------------------------------------------------------------
```

```
PACKET   localmanager  :


create ........ (TEXT C name)
new ........... (TEXT C name) --> DATASPACE
old ........... (TEXT C name) --> DATASPACE
old ........... (TEXT C name, INT C expectedtype) --> DATASPACE
exists ........ (TEXT C name) --> BOOL
forget ........ (TEXT C name)
forget ........
status ........ (TEXT C name, statustext)
status ........ (TEXT C name) --> TEXT
status ........ (INT C pos, TEXT C statuspattern)
copy .......... (DATASPACE C source, TEXT C destname)
copy .......... (TEXT C sourcename, destname)
rename ........ (TEXT C oldname, newname)
beginlist .....
getlistentry .. (TEXT V entry, statustext)
killer ........
list ..........
list .......... (FILE V f)
filenames ..... (FILE V f)
password ...... (TEXT C name) --> TEXT
password ...... --> TEXT
enterpassword . (TEXT C password)
enterpassword . (TEXT C filename, password)
readpermission  (TEXT C name, supplypassword) --> BOOL
writepermission  (TEXT C name, supplypassword) --> BOOL


possible errors :file already existing
                 create impossible
                 file does not exist
                 dataspace has wrong type
                 destination file already existing
-------------------------------------------------------------------------


PACKET   file  :


FILE .......... TYPE
input ......... --> TRANSPUTDIRECTION
output ........ --> TRANSPUTDIRECTION
modify ........ --> TRANSPUTDIRECTION
sequentialfile  (TRANSPUTDIRECTION C mode) --> FILE
sequentialfile  (TRANSPUTDIRECTION C mode, DATASPACE V ds) --> FILE
sequentialfile  (TRANSPUTDIRECTION C mode, TEXT C name) --> FILE
maxlinelength . (FILE C file) --> INT
maxlinelength . (FILE V file, INT C length)
headline ...... (FILE V file, TEXT C head)
headline ...... (FILE V file) --> TEXT
copyattributes  (FILE C source, FILE V dest)
input ......... (FILE V file)
```

```
output ........ (FILE V file)
modify ........ (FILE V file)
putline ....... (FILE V file, TEXT C record)
getline ....... (FILE V file, TEXT V record)
line .......... (FILE V file)
reset ......... (FILE V file)
eof ........... (FILE C file) --> BOOL
line .......... (FILE V file, INT C lines)
page .......... (FILE V file)
eop ........... (FILE C file) --> BOOL
put ........... (FILE V file, TEXT C word)
put ........... (FILE V f, INT C value)
put ........... (FILE V f, REAL C real)
out ........... (FILE V file, TEXT C word)
get ........... (FILE V file, TEXT V word, TEXT C separator)
get ........... (FILE V file, TEXT V word, INT C maxlength)
get ........... (FILE V file, TEXT V word)
get ........... (FILE V f, INT V number)
get ........... (FILE V f, REAL V number)
close ......... (FILE V file)
close .........
maxpagelength . (FILE C file) --> INT
maxpagelength . (FILE V file, INT C length)
readrecord .... (FILE C file, TEXT V record)
writerecord ... (FILE V file, TEXT C record)
forward ....... (FILE V file)
backward ...... (FILE V file)
deleterecord .. (FILE V file)
insertrecord .. (FILE V file)
tofirstrecord . (FILE V file)
toeof ......... (FILE V file)
isfirstrecord . (FILE C file) --> BOOL
reorganize .... (TEXT C filename)
reorganize ....
feldout ....... (FILE C file, TEXT C satz)
feldeinruecken  (FILE C file, TEXT C satz)
feldeditor .... (FILE V file, TEXT C satz)
pos ........... (FILE C file, TEXT C pattern, INT C from) --> INT
change ........ (FILE V file, INT C from, to, TEXT C new)
subtext ....... (FILE C file, INT C from) --> TEXT
subtext ....... (FILE C file, INT C from, to) --> TEXT
sort .......... (TEXT C dateiname)
sort .......... (TEXT C dateiname, INT C sortieranfang)
sort .......... (TEXT C dateiname, feldname)
:= ............ (FILE V a, FILE C b)
```

```
possible errors :dataspace has wrong type
                 input file not existing
                 file not open
                 operation not in transputdirection 'modify'
                 operation only in transputdirection 'modify'
                 input after end of file
                 input access to output file
                 output access to input file
                 forward at eof
                 backward at first record
                 file does not exist
                 sort: feldname
-------------------------------------------------------------------------


PACKET   scanner  :


scan ..... (TEXT C scantext)
continuescan .. (TEXT C scantext)
fixscanner ....
resetscanner ..
nextsymbol .... (TEXT V symbol)
nextsymbol .... (TEXT V symbol, INT V type)
-------------------------------------------------------------------------


PACKET   dateieditorpaket  :


dateieditor ... (DATEI V datei)
defineescape .. (TEXT C cmdchar, kommando)
-------------------------------------------------------------------------


PACKET   editor  :


edit ..... (FILE V file)
edit ..... (FILE V file1, file2)
edit ..... (TEXT C filename)
edit .....
edit ..... (TEXT C filename1, filename2)
show ..... (FILE V file)
show ..... (TEXT C filename)
editmode ......


possible errors : file does not exist
-------------------------------------------------------------------------
```

```
PACKET   mathlib  :


pi ............ --> REAL
e ............. --> REAL
ln ............ (REAL C x) --> REAL
log2 .......... (REAL C z) --> REAL
log10 ......... (REAL C x) --> REAL
sqrt .......... (REAL C z) --> REAL
exp ........... (REAL C z) --> REAL
tan ........... (REAL C x) --> REAL
tand .......... (REAL C x) --> REAL
sin ........... (REAL C x) --> REAL
sind .......... (REAL C x) --> REAL
cos ........... (REAL C x) --> REAL
cosd .......... (REAL C x) --> REAL
arctan ........ (REAL C x) --> REAL
arctand ....... (REAL C x) --> REAL
** ............ (REAL C base, exponent) --> REAL
** ............ (REAL C a, INT C b) --> REAL
random ........ --> REAL
initializerandom (REAL C z)


possible errors :log2 mit negativer zahl
                 sqrt von negativer zahl
                 hoch mit negativer basis
                 0**0 geht nicht
-------------------------------------------------------------------------



PACKET   elan  :


do ............ (TEXT C command)
run ........... (TEXT C filename)
run ............
runagain .......
insert ........ (TEXT C filename)
insert .........
prot .......... (TEXT C protfilename)
protoff ........
checkon ........
checkoff .......


possible errors : run again impossible
-------------------------------------------------------------------------
```

```
PACKET   commandhandler  :


setcommand .... (TEXT C command, INT C type)
docommand .....
commandhandler  (TEXT C commandlist, INT V commandindex, numberofparams,
                 TEXT V param1, param2, TEXT C commandtext)
commandhandler  (TEXT C commandlist, INT V commandindex, numberofparams,
                 TEXT V param1, param2)
commanderror ..


possible errors :incorrect procedure name
                 ) expected
                 ( expected
                 command too complex
                 parameter is no text denoter (" missing!)
-------------------------------------------------------------------------


PACKET   iochannels  :


channel ....... --> INT
channel ....... (INT C channelnr)
channelexists . (INT C channelnr) --> BOOL
eumelmustadvertise
-------------------------------------------------------------------------


PACKET   tasks  :

                                                                      (M)
TASK .......... TYPE                                                  (M)
syscat ........ --> DATASPACE                                         (M)
myself ........ --> TASK                                              (M)
:= ............ (TASK V dest, TASK C source)                         (M)
= ............. (TASK C left, right) --> BOOL                        (M)
isniltask ..... (TASK C id) --> BOOL                                  (M)
< ............. (TASK C left, right) --> BOOL                        (M)
exists ........ (TASK C id) --> BOOL                                  (M)
name .......... (TASK C id) --> TEXT                                  (M)
access ........ (TASK C id)                                           (M)
task .......... (TEXT C taskname) --> TASK                            (M)
index ......... (TASK C id) --> INT                                   (M)
father ........ --> TASK                                              (M)
father ........ (TASK C id) --> TASK                                  (M)
son ........... (TASK C id) --> TASK                                  (M)
brother ....... (TASK C id) --> TASK                                  (M)
accesscatalogue                                                       (M)
entry ......... (TASK C fathertask, TEXT C taskname, TASK V newtask) (M)
entry ......... (TASK C fathertask, TASK V newtask)                   (M)
delete ........ (TASK C superfluous)                                  (M)
```

```
send ..... (TASK C dest, INT C sendcode, DATASPACE V ds,
            INT V quit)                                          (M)
send ..... (TASK C dest, INT C sendcodeDATASPACE V ds)           (M)
wait ..... (DATASPACE V ds, INT V receivecode, TASK V source)(M)
call ..... (TASK C dest, INT C ordercode, DATASPACE V ds,
            INT V replycode)                                     (M)
begin ..... (TEXT C sonname, PROCstart)                          (M)
begin ..... (DATASPACE V ds, PROCstart, INT V reply)             (M)
disablebegin ..                                                  (M)
end .....                                                        (M)
end ..... (TASK C id)                                            (M)
break .....                                                      (M)
continue ..... (INT C channelnr)                                 (M)
taskpassword .. (TEXT C password)                                (M)
setautonom ....                                                  (M)
resetautonom ..                                                  (M)


possible errors :task not existing
                 incorrect task name
                 duplicate task name
                 father task not existing
                 task directory overflow
                 begin disabled
-----------------------------------------------------------------------


PACKET   namedcommunication  :

call ..... (TEXT C destname, INT C ordercode, DATASPACE V ds,
            INT V replycode)                                     (M)
-----------------------------------------------------------------------


PACKET   taskpcbreadaccess  :

pcb ..... (TASK C id, INT C field) --> INT                       (M)
status ..... (TASK C id) --> INT                                 (M)
channel ..... (TASK C id) --> INT                                (M)
clock ..... (TASK C id) --> REAL                                 (M)
-----------------------------------------------------------------------
```

```
PACKET   globalmanager  :

fetch ......... (TEXT C filename)                                    (M)
fetch ......... (TEXT C filename, fathername)                        (M)
save ..........                                                      (M)
save ..........  (TEXT C filename)                                   (M)
save ..........  (TEXT C filename, fathername)                       (M)
read ..........  (TEXT C filename)                                   (M)
read ..........  (TEXT C filename, fathername)                       (M)
erase .........                                                      (M)
erase .........  (TEXT C filename)                                   (M)
erase .........  (TEXT C filename, fathername)                       (M)
list ..........  (TEXT C fathername)                                 (M)
list ..........  (FILE V f, TEXT C fathername)                       (M)
filenames .....  (FILE V f, TEXT C fathername)                       (M)
globalmanager .                                                      (M)

possible errors :not parent
                 wrong operation
                 wrong password
-------------------------------------------------------------------------


PACKET   printcmd  :

print .........
print .........  (TEXT C filename)
-------------------------------------------------------------------------


PACKET   systeminfo  :

taskinfo ......                                                      (M)
-------------------------------------------------------------------------
```

```
PACKET   archivesystem  :


archive ....... (TEXT C name)
releasearchive
toarchive ..... (TEXT C filename)
toarchive .....
fromarchive ... (TEXT C filename)
cleararchive ..
listarchive ...
listarchive ... (FILE V listfile)
archivenames .. (FILE V listfile)
savearchive ...
savearchive ... (TEXT C filename)
savearchive ... (FILE V dir)
loadarchive ...


possible errors :wrong archive: 1.5
                 wrong archive:
                 file does not exist
                 file not existing in archive
                 archive read/write impossible
                 archive read/write error
                 archive overflow
                 rerun during archive access
                 wrong archive format
                 not enough system storage
-------------------------------------------------------------------------


PACKET   monitor  :


singleusermonitor                                                    (S)
multiusermonitor                                                     (M)
storageinfo ...                                                      (M)
storage ....... (INT V size, used)                                   (M)
help ..........
-------------------------------------------------------------------------
```

```
PACKET   vector  :


VECTOR ........ TYPE                                                  (+)
:= ............ (VECTOR V l, VECTOR C r)                             (+)
:= ............ (VECTOR V l, INITVECTOR C r)                         (+)
nilvector ..... --> INITVECTOR                                       (+)
vector ........ (INT C lng, REAL C value) --> INITVECTOR             (+)
vector ........ (INT C lng) --> INITVECTOR                           (+)
SUB ........... (VECTOR C v, INT C i) --> REAL                       (+)
LENGTH ........ (VECTOR C v) --> INT                                 (+)
length ........ (VECTOR C v) --> INT                                 (+)
norm .......... (VECTOR C v) --> REAL                                (+)
replace ....... (VECTOR V v, INT C i, REAL C r)                      ( )
= ............. (VECTOR C l, r) --> BOOL                            (+)
<> ............ (VECTOR C l, r) --> BOOL                            (+)
+ ............. (VECTOR C v) --> VECTOR                             (+)
+ ............. (VECTOR C l, r) --> VECTOR                          (+)
- ............. (VECTOR C a) --> VECTOR                             (+)
- ............. (VECTOR C l, r) --> VECTOR                          (+)
* ............. (VECTOR C l, r) --> REAL                            (+)
* ............. (VECTOR C v, REAL C r) --> VECTOR                   (+)
* ............. (REAL C r, VECTOR C a) --> VECTOR                   (+)
/ ............. (VECTOR C a, REAL C r) --> VECTOR                   (+)
get ........... (VECTOR V v, INT C lng)                              (+)
put ........... (VECTOR C v, INT C length, fracs)                    (+)
put ........... (VECTOR C v)                                         (+)


possible errors :PROC vector : lng <= 0
                 subscript overflow
                 subscript underflow
                 undefined
-----------------------------------------------------------------------


PACKET   matrix  :


MATRIX ........ TYPE                                                  (+)
:= ............ (MATRIX V l, MATRIX C r)                             (+)
:= ............ (MATRIX V l, INITMATRIX C r)                         (+)
matrix ........ (INT C rows, columns, REAL C value) --> INITMATRIX   (+)
matrix ........ (INT C rows, columns) --> INITMATRIX                 (+)
idn ........... (INT C size) --> INITMATRIX                          (+)
row ........... (MATRIX C m, INT C i) --> VECTOR                     (+)
column ........ (MATRIX C m, INT C j) --> VECTOR                     (+)
COLUMNS ....... (MATRIX C m) --> INT                                 (+)
ROWS .......... (MATRIX C m) --> INT                                 (+)
sub ........... (MATRIX C a, INT C row, column) --> REAL             (+)
replacerow .... (MATRIX V m, INT C rowindex, VECTOR C rowvalue)      (+)
replacecolumn . (MATRIX V m,INT C columnindex, VECTOR C columnvalue) (+)
replaceelement  (MATRIX V a, INT C row, column, REAL C r)            (+)
= ............. (MATRIX C l, r) --> BOOL                            (+)
```

```
<> ..... (MATRIX C l, r) --> BOOL                                    (+)
+ ..... (MATRIX C m) --> MATRIX                                      (+)
+ ..... (MATRIX C l, r) --> MATRIX                                   (+)
- ..... (MATRIX C m) --> MATRIX                                      (+)
- ..... (MATRIX C l, r) --> MATRIX                                   (+)
* ..... (REAL C x, MATRIX C m) --> MATRIX                           (+)
* ..... (MATRIX C m, REAL C x) --> MATRIX                           (+)
* ..... (VECTOR C v, MATRIX C m) --> VECTOR                         (+)
* ..... (MATRIX C m, VECTOR C v) --> VECTOR                         (+)
* ..... (MATRIX C l, r) --> MATRIX                                  (+)
get ..... (MATRIX V a, INT C rows, columns)                         (+)
put ..... (MATRIX C a, INT C length, fracs)                         (+)
put ..... (MATRIX C a)                                              (+)
TRANSP ..... (MATRIX C m) --> MATRIX                                (+)
transp ..... (MATRIX V m)                                           (+)
INV ..... (MATRIX C m) --> MATRIX                                   (+)
DET ..... (MATRIX C m) --> REAL                                     (+)


possible errors :PROC matrix : rows <= 0
                 PROC matrix : columns <= 0
                 MATRIX PROC idn : size <= 0
                 subscript underflow
                 subscript overflow
                 undefined
                 MATRIX OP INV : no square matrix
                 MATRIX OP INV : singular matrix
                 REAL OP DET : no square matrix
-------------------------------------------------------------------------


PACKET   complex  :


COMPLEX ..... TYPE                                                  (+)
complexzero ... --> COMPLEX                                         (+)
complexone .... --> COMPLEX                                         (+)
complexi ..... --> COMPLEX                                          (+)
:= ..... (COMPLEX V dest, COMPLEX C source)                         (+)
complex ..... (REAL C re, im) --> COMPLEX                           (+)
realpart ..... (COMPLEX C number) --> REAL                          (+)
imagpart ..... (COMPLEX C number) --> REAL                          (+)
CONJ ..... (COMPLEX C number) --> COMPLEX                           (+)
= ..... (COMPLEX C a, b) --> BOOL                                   (+)
<> ..... (COMPLEX C a, b) --> BOOL                                  (+)
+ ..... (COMPLEX C a, b) --> COMPLEX                                (+)
- ..... (COMPLEX C a, b) --> COMPLEX                                (+)
* ..... (COMPLEX C a, b) --> COMPLEX                                (+)
/ ..... (COMPLEX C a, b) --> COMPLEX                                (+)
```

```
get ..... (COMPLEX V a)                                          (+)
put ..... (COMPLEX C a)                                          (+)
dphi ..... (COMPLEX C x) --> REAL                                (+)
phi ..... (COMPLEX C x) --> REAL                                 (+)
dphi ..... (COMPLEX C x) --> REAL                                (+)
phi ..... (COMPLEX C x) --> REAL                                 (+)
sqrt ..... (COMPLEX C x) --> COMPLEX                             (+)
ABS ..... (COMPLEX C x) --> REAL                                 (+)
-----------------------------------------------------------------------
```

```
PACKET   longint  :
```

```
LONGINT ..... TYPE                                               (+)
:= ..... (LONGINT V left, LONGINT C right)                       (+)
< ..... (LONGINT C left, right) --> BOOL                         (+)
> ..... (LONGINT C left, right) --> BOOL                         (+)
<= ..... (LONGINT C left, right) --> BOOL                        (+)
>= ..... (LONGINT C left, right) --> BOOL                        (+)
<> ..... (LONGINT C left, right) --> BOOL                        (+)
= ..... (LONGINT C left, right) --> BOOL                         (+)
- ..... (LONGINT C arg) --> LONGINT                              (+)
+ ..... (LONGINT C arg) --> LONGINT                              (+)
- ..... (LONGINT C left, right) --> LONGINT                      (+)
+ ..... (LONGINT C left, right) --> LONGINT                      (+)
* ..... (LONGINT C left, right) --> LONGINT                      (+)
** ..... (LONGINT C arg, exp) --> LONGINT                        (+)
** ..... (LONGINT C arg, INT C exp) --> LONGINT                  (+)
ABS ..... (LONGINT C arg) --> LONGINT                            (+)
abs ..... (LONGINT C a) --> LONGINT                              (+)
DECR ..... (LONGINT V result, LONGINT C ab)                      (+)
DIV ..... (LONGINT C left, right) --> LONGINT                    (+)
get ..... (LONGINT V result)                                     (+)
get ..... (FILE V file, LONGINT V result)                        (+)
INCR ..... (LONGINT V result, LONGINT C dazu)                    (+)
int ..... (LONGINT C longint) --> INT                            (+)
longint ..... (INT C int) --> LONGINT                            (+)
longint ..... (TEXT C text) --> LONGINT                          (+)
max ..... (LONGINT C left, right) --> LONGINT                    (+)
maxlongint .... --> LONGINT                                      (+)
min ..... (LONGINT C left, right) --> LONGINT                    (+)
MOD ..... (LONGINT C left, right) --> LONGINT                    (+)
put ..... (LONGINT C longint)                                    (+)
put ..... (FILE V file, LONGINT C longint)                       (+)
random ..... (LONGINT C lowerbound, upperbound) --> LONGINT      (+)
SIGN ..... (LONGINT C arg) --> INT                               (+)
sign ..... (LONGINT C arg) --> INT                               (+)
text ..... (LONGINT C longint) --> TEXT                          (+)
text ..... (LONGINT C longint, INT C length) --> TEXT            (+)
zero ..... --> LONGINT                                           (+)
```

```
possible errors :LONGINT OP ** : negativ exp
                 LONGINT OP DIV by zero
                 LONGINT OP MOD by zero
-----------------------------------------------------------------------


PACKET   picture  :

PICFILE ....... TYPE                                                  (+)
:= ............ (PICFILE V p, DATASPACE C d)                         (+)
picturefile ... (TEXT C name) --> DATASPACE                           (+)
move .......... (PICFILE V p, REAL C x, y)                            (+)
move .......... (PICFILE V p, REAL C x, y, z)                         (+)
draw .......... (PICFILE V p, REAL C x, y)                            (+)
draw .......... (PICFILE V p, REAL C x, y, z)                         (+)
where ......... (PICFILE C p;REAL V x, y, z)                          (+)
where ......... (PICFILE C p;REAL V x, y)                             (+)
draw .......... (PICFILE V p, TEXT C text)                            (+)
draw .......... (PICFILE V p, TEXT C text, REAL C angle, height)      (+)
pen ........... (PICFILE V p, INT C pennumber)                        (+)
pen ........... (PICFILE V p) --> INT                                 (+)
extrema ....... (PICFILE V p, REAL V xmin, xmax, ymin, ymax)          (+)
extrema ....... (PICFILE V p,
                 REAL V xmin, xmax, ymin, ymax, zmin, zmax)           (+)
translate ..... (PICFILE V p;REAL C dx, dy)                           (+)
translate ..... (PICFILE V p, REAL C dx, dy, dz)                      (+)
stretch ....... (PICFILE V p, REAL C xc, yc)                          (+)
stretch ....... (PICFILE V p, REAL C xc, yc, zc)                      (+)
rotate ........ (PICFILE V p, REAL C angle)                           (+)
rotate ........ (PICFILE V p, REAL C alpha, phi, teta)                (+)
selectpen ..... (PICFILE V p,
                 INT C pen, colour, thickness, linetype)              (+)
view .......... (PICFILE V p;REAL C phi, teta)                        (+)
orthographic .. (PICFILE V p)                                         (+)
oblique ....... (PICFILE V p;REAL C a, b)                             (+)
perspective ... (PICFILE V p;REAL C cx, cy, cz)                       (+)
twodimensional  (PICFILE V p)                                         (+)
window ........ (PICFILE V p;REAL C xmin, xmax, ymin, ymax)           (+)
window ........ (PICFILE V p,
                 REAL C xmin, xmax, ymin, ymax, zmin, zmax)           (+)
viewport ...... (PICFILE V p,
                 REAL C hormin, hormax, vertmin, vertmax)             (+)
plot .......... (PICFILE V p)                                         (+)
CAT ........... (PICFILE V dest, PICFILE C source)                    (+)
```

```
possible errors :dataspace is no PICFILE
                 picfile overflow
                 sequence error
                 wrong picture code (
                 pen < 0 is not possible
                 pen > 16 is not possible
                 pen < 1 is not possible
-------------------------------------------------------------------------


PACKET   spoolmanager  :

spoolmanager ..                                                    (+M)

possible errors :not parent
                 spool overflow
                 your file does not exist
-------------------------------------------------------------------------


PACKET   eumelprinter  :

print ......... (FILE V f, INT C firstpage, lastpage)              (+)
resetprinter ..                                                    (+)
resetprinter .. (BOOL C typeofpaper, INT C paperlen, paperwid)     (+)
printline ..... (TEXT C in)                                        (+)
-------------------------------------------------------------------------


PACKET   zeilenformatieren  :

lineform ......                                                    (+)
lineform ...... (TEXT C dateiname)                                 (+)

possible errors :file does not exists
                 halt from terminal
-------------------------------------------------------------------------
```

```
PACKET   seitenformatierung  :

pageform ..... (TEXT C eingabe, druck)                          (+)
pageform ..... (TEXT C eingabe)                                 (+)
pageform .....                                                  (+)


possible errors :file does not exists
                 input file name = output file name
                 output file already exists
                 too much lines in 'foot'
                 too much lines in 'head' or 'bottom'
                 halt from terminal
-------------------------------------------------------------------


PACKET   cryptograf  :

crypt ......... (TEXT C file, key)                              (+)
decrypt ....... (TEXT C file, key)                              (+)
-------------------------------------------------------------------
```

Bulletin (alphabetisch):

In dieser Liste sind alle verfügbaren Prozeduren, Operatoren und Datentypen des EUMEL-Systems aufgeführt, einschließlich der Prozeduren von System-Programmen und Prozeduren, die verübergehend bzw. versuchsweise in das System übernommen wurden. Die letzteren Prozeduren sind -außer an dieser Stelle - nicht näher beschrieben und können ohne weiteres jederzeit geändert werden. Benutzung erfolgt also auf eigene Gefahr.

Die mit
    (M) sind nur im Multi-User, während die mit
    (+) gekennzeichneten Objekte nicht standardmäßig vorübersetzt sind.
Dabei bedeuten bei den formalen Parametern
    C CONST und
    V VAR.

```
* ..... (COMPLEX C a, b) --> COMPLEX                    (+)
* ..... (INT C left, right) --> INT
* ..... (INT C times, TEXT C source) --> TEXT
* ..... (LONGINT C left, right) --> LONGINT            (+)
* ..... (MATRIX C l, r) --> MATRIX                     (+)
* ..... (MATRIX C m, REAL C x) --> MATRIX              (+)
* ..... (MATRIX C m, VECTOR C v) --> VECTOR            (+)
* ..... (REAL C left, right) --> REAL
* ..... (REAL C x, MATRIX C m) --> MATRIX              (+)
* ..... (VECTOR C l, r) --> REAL                       (+)
* ..... (VECTOR C v, MATRIX C m) --> VECTOR            (+)
* ..... (VECTOR C v, REAL C r) --> VECTOR              (+)
* ..... (REAL C r, VECTOR C a) --> VECTOR              (+)
** ..... (INT C arg, exp) --> INT
** ..... (LONGINT C arg, INT C exp) --> LONGINT        (+)
** ..... (LONGINT C arg, exp) --> LONGINT             (+)
** ..... (REAL C a, INT C b) --> REAL
** ..... (REAL C base, exponent) --> REAL
+ ..... (COMPLEX C a, b) --> COMPLEX                   (+)
+ ..... (INT C left, right) --> INT
+ ..... (LONGINT C arg) --> LONGINT                    (+)
+ ..... (LONGINT C left, right) --> LONGINT            (+)
+ ..... (MATRIX C l, r) --> MATRIX                     (+)
+ ..... (MATRIX C m) --> MATRIX                        (+)
+ ..... (REAL C left, right) --> REAL
+ ..... (TEXT C left, right) --> TEXT
+ ..... (VECTOR C l, r) --> VECTOR                     (+)
+ ..... (VECTOR C v) --> VECTOR                        (+)
```

```
- ..... (COMPLEX C a, b) --> COMPLEX                              (+)
- ..... (INT C left, right) --> INT
- ..... (INT C operand) --> INT
- ..... (LONGINT C arg) --> LONGINT                               (+)
- ..... (LONGINT C left, right) --> LONGINT                       (+)
- ..... (MATRIX C l, r) --> MATRIX                                (+)
- ..... (MATRIX C m) --> MATRIX                                   (+)
- ..... (REAL C left, right) --> REAL
- ..... (REAL C operand) --> REAL
- ..... (VECTOR C a) --> VECTOR                                   (+)
- ..... (VECTOR C l, r) --> VECTOR                                (+)
/ ..... (COMPLEX C a, b) --> COMPLEX                             (+)
/ ..... (REAL C left, right) --> REAL
/ ..... (VECTOR C a, REAL C r) --> VECTOR                        (+)
:= ..... (BOOL V dest, BOOL C source)
:= ..... (COMPLEX V dest, COMPLEX C source)                      (+)
:= ..... (DATASPACE V dest, DATASPACE C source)
:= ..... (FILE V a, FILE C b)
:= ..... (INT V dest, INT C source)
:= ..... (LONGINT V left, LONGINT C right)                       (+)
:= ..... (MATRIX V l, INITMATRIX C r)                            (+)
:= ..... (MATRIX V l, MATRIX C r)                                (+)
:= ..... (PICFILE V p, DATASPACE C d)                            (+)
:= ..... (REAL V dest, REAL C source)
:= ..... (TASK V dest, TASK C source)                            (M)
:= ..... (TEXT V dest, TEXT C source)
:= ..... (THESAURUS V dest, INITIALTHESAURUS C dummy)
:= ..... (VECTOR V l, INITVECTOR C r)                            (+)
:= ..... (VECTOR V l, VECTOR C r)                                (+)
< ..... (INT C left, right) --> BOOL
< ..... (LONGINT C left, right) --> BOOL                         (+)
< ..... (REAL C left, right) --> BOOL
< ..... (TASK C left, right) --> BOOL                            (M)
< ..... (TEXT C left, right) --> BOOL
<= ..... (INT C left, right) --> BOOL
<= ..... (LONGINT C left, right) --> BOOL                        (+)
<= ..... (REAL C left, right) --> BOOL
<= ..... (TEXT C left, right) --> BOOL
<> ..... (COMPLEX C a, b) --> BOOL                               (+)
<> ..... (INT C left, right) --> BOOL
<> ..... (LONGINT C left, right) --> BOOL                        (+)
<> ..... (MATRIX C l, r) --> BOOL                                (+)
<> ..... (REAL C left, right) --> BOOL
<> ..... (TEXT C left, right) --> BOOL
<> ..... (VECTOR C l, r) --> BOOL                                (+)
= ..... (COMPLEX C a, b) --> BOOL                                (+)
= ..... (INT C left, right) --> BOOL
= ..... (LONGINT C left, right) --> BOOL                         (+)
= ..... (MATRIX C l, r) --> BOOL                                 (+)
= ..... (REAL C left, right) --> BOOL
= ..... (TASK C left, right) --> BOOL                            (M)
= ..... (TEXT C left, right) --> BOOL
= ..... (VECTOR C l, r) --> BOOL                                 (+)
```

```
> ..... (INT C left, right) --> BOOL
> ..... (LONGINT C left, right) --> BOOL                    (+)
> ..... (REAL C left, right) --> BOOL
> ..... (TEXT C left, right) --> BOOL
>= ..... (TEXT C left, right) --> BOOL
>= ..... (INT C left, right) --> BOOL
>= ..... (LONGINT C left, right) --> BOOL                  (+)
>= ..... (REAL C left, right) --> BOOL
ABS ..... (COMPLEX C x) --> REAL                           (+)
ABS ..... (INT C argument) --> INT
ABS ..... (LONGINT C arg) --> LONGINT                      (+)
ABS ..... (REAL C value) --> REAL
AND ..... (BOOL C left, right) --> BOOL
BOOL ..... TYPE
CAND ..... (BOOL C left, right) --> BOOL
CAT ..... (PICFILE V dest, PICFILE C source)               (+)
CAT ..... (TEXT V right, TEXT C left)
COLUMNS ..... (MATRIX C m) --> INT                         (+)
COMPLEX ..... TYPE                                         (+)
CONJ ..... (COMPLEX C number) --> COMPLEX                  (+)
CONTAINS ..... (THESAURUS V thesaurus, TEXT C name) --> BOOL
COR ..... (BOOL C left, right) --> BOOL
DECR ..... (INT V dest, INT C decrement)
DECR ..... (LONGINT V result, LONGINT C ab)                (+)
DECR ..... (REAL V dest, REAL C decrement)
DET ..... (MATRIX C m) --> REAL                            (+)
DIV ..... (INT C left, right) --> INT
DIV ..... (LONGINT C left, right) --> LONGINT              (+)
FILE ..... TYPE
INCR ..... (INT V dest, INT C increment)
INCR ..... (LONGINT V result, LONGINT C dazu)              (+)
INCR ..... (REAL V dest, REAL C increment)
INITFLAG ..... TYPE
INT ..... TYPE
INV ..... (MATRIX C m) --> MATRIX                          (+)
ISUB ..... (TEXT C text, INT C index) --> INT
LENGTH ..... (VECTOR C v) --> INT                          (+)
LENGTH ..... (TEXT C text) --> INT
LONGINT ..... TYPE                                         (+)
MATRIX ..... TYPE                                          (+)
MOD ..... (INT C left, right) --> INT
MOD ..... (LONGINT C left, right) --> LONGINT              (+)
MOD ..... (REAL C left, right) --> REAL
NOT ..... (BOOL C operand) --> BOOL
OR ..... (BOOL C left, right) --> BOOL
PICFILE ..... TYPE                                         (+)
REAL ..... TYPE
ROWS ..... (MATRIX C m) --> INT                            (+)
RSUB ..... (TEXT C text, INT C index) --> REAL
SIGN ..... (INT C argument) --> INT
SIGN ..... (LONGINT C arg) --> INT                         (+)
SIGN ..... (REAL C value) --> INT
SUB ..... (TEXT C text, INT C pos) --> TEXT
SUB ..... (VECTOR C v, INT C i) --> REAL                   (+)
```

```
TASK ..... TYPE                                                    (M)
TEXT ..... TYPE
THESAURUS ..... TYPE
TIMESOUT ..... (INT C times, TEXT C text)
TRANSP ..... (MATRIX C m) --> MATRIX                               (+)
VECTOR ..... TYPE                                                  (+)
XOR ..... (BOOL C left, right) --> BOOL
abs ..... (INT C argument) --> INT
abs ..... (LONGINT C a) --> LONGINT                                (+)
abs ..... (REAL C value) --> REAL
access ..... (TASK C id)                                           (M)
accesscatalogue                                                    (M)
archive ..... (TEXT C name)
archivenames .. (FILE V listfile)
arctan ..... (REAL C x) --> REAL
arctand ..... (REAL C x) --> REAL
backward ..... (FILE V file)
begin ..... (DATASPACE V ds, PROCstart, INT V reply)               (M)
begin ..... (TEXT C sonname, PROCstart)                            (M)
beginlist .....
break .....                                                        (M)
brother ..... (TASK C id) --> TASK                                 (M)
call ..... (TASK C dest, INT C ordercode, DATASPACE V ds,
             INT V replycode)                                      (M)
call ..... (TEXT C destname, INT C ordercode, DATASPACE V ds,
             INT V replycode)                                      (M)
change ..... (FILE V file, INT C from, to, TEXT C new)
change ..... (TEXT V destination, INT C from, to, TEXT C new)
change ..... (TEXT V destination, TEXT C old, new)
changeall ..... (TEXT V destination, TEXT C old, new)
channel ..... (INT C channelnr)
channel ..... (TASK C id) --> INT                                  (M)
channel ..... --> INT
channelexists . (INT C channelnr) --> BOOL
checkoff .....
checkon .....
cleararchive ..
clearerror ....
clock ..... (INT C nr) --> REAL
clock ..... (TASK C id) --> REAL                                   (M)
close .....
close ..... (FILE V file)
code ..... (INT C code) --> TEXT
code ..... (TEXT C text) --> INT
column ..... (MATRIX C m, INT C j) --> VECTOR                      (+)
commanddialogue  (BOOL C status)
commanddialogue  --> BOOL
commanderror ..
commandhandler  (TEXT C commandlist, INT V commandindex, numberofparams,
                 TEXT V param1, param2)
commandhandler  (TEXT C commandlist, INT V commandindex, numberofparams,
                 TEXT V param1, param2, TEXT C commandtext)
```

```
complex ..... (REAL C re, im) --> COMPLEX                          (+)
complexi ..... --> COMPLEX                                         (+)
complexone .... --> COMPLEX                                        (+)
complexzero ... --> COMPLEX                                        (+)
compress ..... (TEXT C text) --> TEXT
continue ..... (INT C channelnr)                                   (M)
continuescan .. (TEXT C scantext)
copy ......... (DATASPACE C source, TEXT C destname)
copy ......... (TEXT C sourcename, destname)
copyattributes  (FILE C source, FILE V dest)
cos .......... (REAL C x) --> REAL
cosd ......... (REAL C x) --> REAL
cout ......... (INT C number)
create ....... (TEXT C name)
crypt ........ (TEXT C file, key)                                  (+)
cursor ....... (INT C x, y)
date ......... (REAL C datum) --> TEXT
date ......... (TEXT C datum) --> REAL
date ......... --> TEXT
dateieditor ... (DATEI V datei)
decrypt ...... (TEXT C file, key)                                  (+)
defineescape .. (TEXT C cmdchar, kommando)
delete ....... (TASK C superfluous)                                (M)
delete ....... (THESAURUS V thesaurus, INT C index)
delete ....... (THESAURUS V thesaurus, TEXT C name, INT V index)
deletechar .... (TEXT V string, INT C deletepos)
deleterecord .. (FILE V file)
disablebegin ..                                                    (M)
disablestop ...
do ........... (TEXT C command)
docommand .....
dphi ......... (COMPLEX C x) --> REAL                              (+)
dphi ......... (COMPLEX C x) --> REAL                              (+)
draw ......... (PICFILE V p, REAL C x, y)                          (+)
draw ......... (PICFILE V p, REAL C x, y, z)                       (+)
draw ......... (PICFILE V p, TEXT C text)                          (+)
draw ......... (PICFILE V p, TEXT C text, REAL C angle, height)    (+)
e ............ --> REAL
edit ..........
edit ......... (FILE V file)
edit ......... (FILE V file1, file2)
edit ......... (TEXT C filename)
edit ......... (TEXT C filename1, filename2)
editget ...... (TEXT V editsatz)
editget ...... (TEXT V editsatz, INT C editlimit, editfende)
editmode ......
emptythesaurus  --> INITIALTHESAURUS
enablestop ....
end ...........                                                    (M)
end ........... (TASK C id)                                        (M)
enterpassword . (TEXT C filename, password)
enterpassword . (TEXT C password)
entry ......... (TASK C fathertask, TASK V newtask)                (M)
entry ......... (TASK C fathertask, TEXT C taskname, TASK V newtask) (M)
```

```
eof ..... (FILE C file) --> BOOL
eop ..... (FILE C file) --> BOOL
erase ..... (M)
erase ..... (TEXT C filename) (M)
erase ..... (TEXT C filename, fathername) (M)
errorline ..... --> INT
errormessage .. --> TEXT
errorstop ..... (TEXT C message)
eumelmustadvertise
exists ..... (TASK C id) --> BOOL (M)
exists ..... (TEXT C name) --> BOOL
exp ..... (REAL C z) --> REAL
extrema ..... (PICFILE V p, REAL V xmin, xmax, ymin, ymax) (+)
extrema ..... (PICFILE V p,
                REAL V xmin, xmax, ymin, ymax, zmin, zmax) (+)
father ..... (TASK C id) --> TASK (M)
father ..... --> TASK (M)
feldeditor .... (FILE V file, TEXT C satz)
feldeinruecken (FILE C file, TEXT C satz)
feldout ..... (FILE C file, TEXT C satz)
fetch ..... (TEXT C filename) (M)
fetch ..... (TEXT C filename, fathername) (M)
filenames ..... (FILE V f)
filenames ..... (FILE V f, TEXT C fathername) (M)
fixscanner ....
floor ..... (REAL C real) --> REAL
forget .....
forget ..... (DATASPACE V dataspace)
forget ..... (TEXT C name)
forward ..... (FILE V file)
frac ..... (REAL C value) --> REAL
fromarchive ... (TEXT C filename)
get ..... (COMPLEX V a) (+)
get ..... (FILE V f, INT V number)
get ..... (FILE V f, REAL V number)
get ..... (FILE V file, LONGINT V result) (+)
get ..... (FILE V file, TEXT V word)
get ..... (FILE V file, TEXT V word, INT C maxlength)
get ..... (FILE V file, TEXT V word, TEXT C separator)
get ..... (INT V number)
get ..... (LONGINT V result) (+)
g+ ..... (MATRIX V a, INT C rows, columns) (+)
get ..... (REAL V value)
get ..... (TEXT V word)
get ..... (TEXT V word, INT C length)
get ..... (VECTOR V v, INT C lng) (+)
get ..... (TEXT V word, TEXT C separator)
getcursor ..... (INT V x, y)
getline ..... (FILE V file, TEXT V record)
getline ..... (TEXT V line)
getlistentry .. (TEXT V entry, statustext)
getsecretline . (TEXT V line)
globalmanager . (M)
```

```
headline ..... (FILE V file) --> TEXT
headline ..... (FILE V file, TEXT C head)
heapsize ..... --> INT
help ..........
highestentry .. (THESAURUS C thesaurus) --> INT
idn ........... (INT C size) --> INITMATRIX                    (+)
imagpart ...... (COMPLEX C number) --> REAL                    (+)
inchar ........ (TEXT V character)
incharety ..... (INT C timelimit) --> TEXT
incharety ..... --> TEXT
index ......... (TASK C id) --> INT                            (M)
initialized ... (INITFLAG V flag) --> BOOL
initializerandom (INT C wert)
initializerandom (REAL C z)
input ......... (FILE V file)
input ......... --> TRANSPUTDIRECTION
insert ........
insert ........ (TEXT C filename)
insert ........ (THESAURUS V thesaurus, TEXT C name, INT V index)
insertchar .... (TEXT V string, TEXT C char, INT C insertpos)
insertrecord .. (FILE V file)
int ........... (LONGINT C longint) --> INT                    (+)
int ........... (REAL C value) --> INT
int ........... (TEXT C number) --> INT
iserror ....... --> BOOL
isfirstrecord . (FILE C file) --> BOOL
isfull ........ (THESAURUS C thesaurus) --> BOOL
isniltask ..... (TASK C id) --> BOOL                           (M)
killer ........
lastconversionok --> BOOL
lastparam ..... (TEXT C new)
lastparam ..... --> TEXT
length ........ (TEXT C text) --> INT
length ........ (VECTOR C v) --> INT                           (+)
line ..........
line .......... (FILE V file)
line .......... (FILE V file, INT C lines)
line .......... (INT C number)
lineform ......                                                (+)
lineform ...... (TEXT C dateiname)                             (+)
link .......... (THESAURUS V thesaurus, TEXT C name) --> INT
list ..........
list .......... (FILE V f)
list .......... (FILE V f, TEXT C fathername)                  (M)
list .......... (TEXT C fathername)                            (M)
listarchive ...
listarchive ... (FILE V listfile)
ln ............ (REAL C x) --> REAL
loadarchive ...
log10 ......... (REAL C x) --> REAL
log2 .......... (REAL C z) --> REAL
longint ....... (INT C int) --> LONGINT                        (+)
longint ....... (TEXT C text) --> LONGINT                      (+)
```

```
matrix ..... (INT C rows, columns) --> INITMATRIX                (+)
matrix ..... (INT C rows, columns, REAL C value) --> INITMATRIX  (+)
max ........ (INT C first, second) --> INT
max ........ (LONGINT C left, right) --> LONGINT                 (+)
max ........ (REAL C a, b) --> REAL
maxint ..... --> INT
maxlinelength . (FILE C file) --> INT
maxlinelength . (FILE V file, INT C length)
maxlongint .... --> LONGINT                                      (+)
maxpagelength . (FILE C file) --> INT
maxpagelength . (FILE V file, INT C length)
maxreal ..... --> REAL
maxtextlength . --> INT
min ........ (INT C first, second) --> INT
min ........ (LONGINT C left, right) --> LONGINT                 (+)
min ........ (REAL C a, b) --> REAL
modify ..... (FILE V file)
modify ..... --> TRANSPUTDIRECTION
move ....... (PICFILE V p, REAL C x, y)                          (+)
move ....... (PICFILE V p, REAL C x, y, z)                       (+)
multiusermonitor                                                 (M)
myself ..... --> TASK                                            (M)
name ....... (TASK C id) --> TEXT                                (M)
name ....... (THESAURUS C thesaurus, INT C index) --> TEXT
new ........ (TEXT C name) --> DATASPACE
nextsymbol .... (TEXT V symbol)
nextsymbol .... (TEXT V symbol, INT V type)
nilspace ..... --> DATASPACE
nilvector ..... --> INITVECTOR                                   (+)
no ......... (TEXT C question) --> BOOL
norm ....... (VECTOR C v) --> REAL                               (+)
oblique ..... (PICFILE V p;REAL C a, b)                          (+)
old ........ (TEXT C name) --> DATASPACE
old ........ (TEXT C name, INT C expectedtype) --> DATASPACE
orthographic .. (PICFILE V p)                                    (+)
out ........ (FILE V file, TEXT C word)
out ........ (TEXT C eins, zwei)
out ........ (TEXT C eins, zwei, drei)
out ........ (TEXT C eins, zwei, drei, vier)
out ........ (TEXT C text)
output ..... (FILE V file)
output ..... --> TRANSPUTDIRECTION
outsubtext .... (TEXT C source, INT C from)
outsubtext .... (TEXT C source, INT C from, to)
page ....... 
page ....... (FILE V file)
pageform ..... (+)
pageform ..... (TEXT C eingabe)                                  (+)
pageform ..... (TEXT C eingabe, druck)                           (+)
paramposition . (INT C x)
password ..... (TEXT C name) --> TEXT
password ..... --> TEXT
pause ...... (INT C timelimit)
pcb ........ (INT C field) --> INT
pcb ........ (TASK C id, INT C field) --> INT                    (M)
```

```
pen ..... (PICFILE V p) --> INT                                    (+)
pen ..... (PICFILE V p, INT C pennumber)                           (+)
perspective ... (PICFILE V p;REAL C cx, cy, cz)                    (+)
phi ..... (COMPLEX C x) --> REAL                                   (+)
phi ..... (COMPLEX C x) --> REAL                                   (+)
pi ..... --> REAL
picturefile ... (TEXT C name) --> DATASPACE                        (+)
plot ..... (PICFILE V p)                                           (+)
pos ..... (FILE C file, TEXT C pattern, INT C from) --> INT
pos ..... (TEXT C source, low, high, INT C from) --> INT
pos ..... (TEXT C source, pattern, INT C from) --> INT
pos ..... (TEXT C source, pattern, INT C from, to) --> INT
pos ..... (TEXT C source, pattern) --> INT
print .....
print ..... (FILE V f, INT C firstpage, lastpage)                  (+)
print ..... (TEXT C filename)
printline ..... (TEXT C in)                                        (+)
prot ..... (TEXT C protfilename)
protoff .....
put ..... (COMPLEX C a)                                            (+)
put ..... (FILE V f, INT C value)
put ..... (FILE V f, REAL C real)
put ..... (FILE V file, LONGINT C longint)                         (+)
put ..... (FILE V file, TEXT C word)
put ..... (INT C number)
put ..... (LONGINT C longint)                                      (+)
put ..... (MATRIX C a, INT C length, fracs)                        (+)
put ..... (MATRIX C a)                                             (+)
put ..... (REAL C real)
put ..... (TEXT C text)
put ..... (VECTOR C v)                                             (+)
put ..... (VECTOR C v, INT C length, fracs)                        (+)
puterror .....
putline ..... (FILE V file, TEXT C record)
putline ..... (TEXT C text)
random ..... (INT C ugrenze, ogrenze) --> INT
random ..... (LONGINT C lowerbound, upperbound) --> LONGINT        (+)
random ..... --> REAL
read ..... (TEXT C filename)                                       (M)
read ..... (TEXT C filename, fathername)                           (M)
readpermission  (TEXT C name, supplypassword) --> BOOL
readrecord .... (FILE C file, TEXT V record)
real ..... (INT C value) --> REAL
real ..... (TEXT C text) --> REAL
realpart ..... (COMPLEX C number) --> REAL                         (+)
releasearchive
rename ..... (TEXT C oldname, newname)
rename ..... (THESAURUS V thesaurus, TEXT C old, new)
reorganize ....
reorganize .... (TEXT C filename)
```

```
replace ..... (TEXT V dest, INT C pos, TEXT C source)
replace ..... (TEXT V text, INT C index, REAL C code)
replace ..... (TEXT V text, INT C index, value)
replace ..... (VECTOR V v, INT C i, REAL C r)                    (+)
replacecolumn . (MATRIX V m,
                 INT C columnindex, VECTOR C columnvalue)        (+)
replaceelement  (MATRIX V a, INT C row, column, REAL C x)        (+)
replacerow .... (MATRIX V m, INT C rowindex, VECTOR C rowvalue)  (+)
reset ......... (FILE V file)
resetautonom .. (M)
resetprinter .. (+)
resetprinter .. (BOOL C typeofpaper, INT C paperlen, paperwid)   (+)
resetscanner ..
rotate ........ (PICFILE V p, REAL C alpha, phi, teta)           (+)
rotate ........ (PICFILE V p, REAL C angle)                      (+)
round ......... (REAL C real, INT C digits) --> REAL
row ........... (MATRIX C m, INT C i) --> VECTOR                 (+)
run ............
run ........... (TEXT C filename)
runagain ......
save ..........                                                  (M)
save .......... (TEXT C filename)                                (M)
save .......... (TEXT C filename, fathername)                    (M)
savearchive ...
savearchive ... (FILE V dir)
savearchive ... (TEXT C filename)
say ........... (TEXT C message)
scan .......... (TEXT C scantext)
selectpen ..... (PICFILE V p,
                 INT C pen, colour, thickness, linetype)         (+)
send .......... (TASK C dest, INT C sendcodeDATASPACE V ds)      (M)
send .......... (TASK C dest,
                 INT C sendcode, DATASPACE V ds, INT V quit)     (M)
sequentialfile  (TRANSPUTDIRECTION C mode) --> FILE
sequentialfile  (TRANSPUTDIRECTION C mode, DATASPACE V ds) --> FILE
sequentialfile  (TRANSPUTDIRECTION C mode, TEXT C name) --> FILE
setautonom .... (M)
setcommand .... (TEXT C command, INT C type)
setconversion . (BOOL C success)
setlinenr ..... (INT C value)
show .......... (FILE V file)
show .......... (TEXT C filename)
sign .......... (INT C argument) --> INT
sign .......... (LONGINT C arg) --> INT                          (+)
sign .......... (REAL C value) --> INT
sin ........... (REAL C x) --> REAL
sind .......... (REAL C x) --> REAL
singleusermonitor                                                (S)
smallreal ..... --> REAL
son ........... (TASK C id) --> TASK                             (M)
sort .......... (TEXT C dateiname)
sort .......... (TEXT C dateiname, INT C sortieranfang)
sort .......... (TEXT C dateiname, feldname)
spoolmanager .. (+M)
```

```
sqrt ..... (COMPLEX C x) --> COMPLEX                              (+)
sqrt ..... (REAL C z) --> REAL
status ..... (INT C pos, TEXT C statuspattern)
status ..... (TASK C id) --> INT                                  (M)
status ..... (TEXT C name) --> TEXT
status ..... (TEXT C name, statustext)
stop .....
storage ..... (INT V size, used)                                  (M)
storageinfo ...                                                   (M)
stretch ..... (PICFILE V p, REAL C xc, yc)                        (+)
stretch ..... (PICFILE V p, REAL C xc, yc, zc)                    (+)
sub ..... (MATRIX C a, INT C row, column) --> REAL                (+)
subtext ..... (FILE C file, INT C from) --> TEXT
subtext ..... (FILE C file, INT C from, to) --> TEXT
subtext ..... (TEXT C source, INT C from) --> TEXT
subtext ..... (TEXT C source, INT C from, to) --> TEXT
syscat ..... --> DATASPACE                                        (M)
tan ..... (REAL C x) --> REAL
tand ..... (REAL C x) --> REAL
task ..... (TEXT C taskname) --> TASK                             (M)
taskinfo .....                                                    (M)
taskpassword .. (TEXT C password)                                 (M)
text ..... (INT C number) --> TEXT
text ..... (INT C number, length) --> TEXT
text ..... (LONGINT C longint) --> TEXT                           (+)
text ..... (LONGINT C longint, INT C length) --> TEXT             (+)
text ..... (REAL C real, INT C length, fracs) --> TEXT
text ..... (REAL C real) --> TEXT
text ..... (TEXT C source, INT C length) --> TEXT
text ..... (TEXT C source, INT C length, from) --> TEXT
time ..... (REAL C value) --> TEXT
time ..... (TEXT C time) --> REAL
timeofday ..... (REAL C value) --> TEXT
timeofday ..... --> TEXT
toarchive .....
toarchive ..... (TEXT C filename)
toeof ..... (FILE V file)
tofirstrecord . (FILE V file)
translate ..... (PICFILE V p, REAL C dx, dy, dz)                  (+)
translate ..... (PICFILE V p;REAL C dx, dy)                       (+)
transp ..... (MATRIX V m)                                         (+)
twodimensional  (PICFILE V p)                                     (+)
type ..... (DATASPACE C ds) --> INT
type ..... (DATASPACE C ds, INT C type)
vector ..... (INT C lng) --> INITVECTOR                           (+)
vector ..... (INT C lng, REAL C value) --> INITVECTOR             (+)
view ..... (PICFILE V p;REAL C phi, teta)                         (+)
viewport ..... (PICFILE V p,
                REAL C hormin, hormax, vertmin, vertmax)          (+)
```

```
wait ..... (DATASPACE V ds, INT V receivecode, TASK V source)(M)
where ..... (PICFILE C p;REAL V x, y)                        (+)
where ..... (PICFILE C p;REAL V x, y, z)                     (+)
window ..... (PICFILE V p;REAL C xmin, xmax, ymin, ymax)     (+)
window ..... (PICFILE V p;
               REAL C xmin, xmax, ymin, ymax, zmin, zmax)    (+)
writepermission  (TEXT C name, supplypassword) --> BOOL
writerecord ... (FILE V file, TEXT C record)
yes ..... (TEXT C question) --> BOOL
zero ..... --> LONGINT                                       (+)
```