Multitasking

das Paket MTASK erlaubt kooperatives Multitaskting.

Laden mit

```
1 GET MTASK
1 5 THRU
```

Glossar

```
(PAUSE)   ( -- )
RESTART   ( -- )
6 CONSTANT INT#
LOCAL   ( base adr -- adr' )
@LINK   ( -- adr )
!LINK   ( adr -- )
SLEEP   ( adr -- )
WAKE    ( adr -- )
STOP    ( -- )
SINGLE   ( -- )
MULTI   ( -- )
TASK:  ( size -- )
SET-TASK   ( IP Task -- )
ACTIVATE   ( Task -- ) Activate a Task
BACKGROUND:   ( -- )   Create a Background task
```

aus L/P-F83-Doku:

```
Multitasking low level
(PAUSE)   (S -- )
    Puts a task to sleep by storing the IP and the RP on the
    parameter stack.  It then saves the pointer to the
    parameter stack in the user area and jumps to the code
    pointed at by USER+3, switching tasks.
RESTART     (S -- )
    Sets the user pointer to point to a new user area and
    restores the parameter stack that was previously saved
    in the USER area.  Then pops the RP and IP off of the
    stack and resumes execution.   The inverse of PAUSE.

Initialize current User area to a single task.

Manipulate Tasks
LOCAL  Map a User variable from the current task to another task
@LINK  Return a pointer the the next tasks entry point
!LINK  Set the link field of the current task (perhaps relative)
SLEEP  makes a task pause indefinitely.
WAKE  lets a task start again.
```

```
STOP  makes a task pause indefinitely.
SINGLE  removes the multi-tasker's scheduler/dispatcher loop.
MULTI
   installs the multi-tasker's scheduler/dispatcher loop.
   By patching the appropriate INT vector and enabling PAUSE.
```

Beispiel

```
VARIABLE COUNTS
BACKGROUND: COUNTER  BEGIN
    PAUSE 1 COUNTS +!  AGAIN ;
COUNTER WAKE  MULTI

COUNTS ?
COUNTS ?
COUNTS ?
SINGLE
```