

# **f90\_unix: UNIX Functions Module**

March 15, 2019

## **1 Name**

`f90_unix` — Module providing access to UNIX functions.

## **2 Usage**

**USE F90\_UNIX**

## **3 Synopsis**

### **Parameters**

`CLOCK_TICK_KIND`

### **Derived Types**

`TMS`

### **Procedures**

`ABORT`, `CLOCK_TICKS_PER_SECOND`, `EXIT`, `FLUSH`, `GETARG`, `GETENV`, `GETGID`,  
`GETPID`, `GETUID`, `IARGC`, `TIMES`

## **4 Parameter Description**

`INTEGER,PARAMETER :: CLOCK_TICK_KIND`

The integer kind used for clock ticks (see `TIMES`).

## **5 Derived-Type Description**

```
TYPE TMS
  INTEGER(CLOCK_TICK_KIND) UTIME, STIME, CUTIME, CTIME
END TYPE
```

Derived type holding CPU usage time in clock ticks. `UTIME` and `STIME` contain CPU time information for a process, `CUTIME` and `CTIME` contain CPU time information for its terminated child processes. In each case this is divided into user time (`UTIME`, `CUTIME`) and system time (`STIME`, `CTIME`).

## **6 Procedure Descriptions**

**SUBROUTINE** `ABORT(message)`

`CHARACTER*(*)`, **OPTIONAL** `:: message`

ABORT cleans up the i/o buffers and then terminates execution, producing a core dump on Unix systems. If MESSAGE is given it is written to logical unit 0 (zero) preceded by ' abort: '.

```
INTEGER (KIND=CLOCK_TICK_KIND) FUNCTION CLOCK_TICKS_PER_SECOND()
```

Returns the number of clock ticks in one second of CPU time (see TIMES).

```
SUBROUTINE EXIT(STATUS)
```

```
INTEGER,OPTIONAL :: STATUS
```

Terminate execution as if executing the END statement of the main program (or an unadorned STOP statement). If STATUS is given it is returned to the operating system (where applicable) as the execution status code.

```
SUBROUTINE FLUSH(LUNIT)
```

```
INTEGER,INTENT(IN) :: LUNIT
```

Flushes the output buffer of logical unit LUNIT which must be connected for formatted sequential output.

```
SUBROUTINE GETARG(K,ARG)
```

```
INTEGER,INTENT(IN) :: K
```

```
CHARACTER*(*),INTENT(OUT) :: ARG
```

Fetches command-line argument number K into ARG. Argument zero is the program name. Note that K must be less than or equal to the value returned by IARGC().

```
SUBROUTINE GETENV(NAME,VALUE)
```

```
CHARACTER*(*),INTENT(IN) :: NAME
```

```
CHARACTER*(*),INTENT(OUT) :: VALUE
```

Fetches the value of the environment variable named by NAME into VALUE. If there is no such variable VALUE will be set to all blanks.

```
INTEGER FUNCTION GETGID()
```

Returns the numeric group number of the calling process.

```
INTEGER FUNCTION GETPID()
```

Returns the process number of the calling process.

```
INTEGER FUNCTION GETUID()
```

Returns the numeric user number of the calling process.

```
INTEGER FUNCTION IARGC()
```

Returns the number of command-line arguments. This will be -1 if even the program name is unavailable.

```
INTEGER(KIND=CLOCK_TICK_KIND) FUNCTION TIMES(BUFFER)
```

```
TYPE(TMS),INTENT(OUT) :: BUFFER
```

This function returns the elapsed real time in clock ticks since an arbitrary point in the past, or -1 if the function is unavailable. `BUFFER` is filled in with CPU time information for the calling process and any terminated child processes.

If this function returns zero the values in `BUFFER` will still be correct but the elapsed-time timer was not available.

## 7 See Also

`nagfor(1)`, `intro(3)`, `nag_modules(3)`.

## 8 Bugs

Please report any bugs found to ‘support@nag.co.uk’ or ‘support@nag.com’, along with any suggestions for improvements.