

**(SYSTEM-SCHEDULER)TERM(c)**

**(SYSTEM-SCHEDULER)TERM(c)**

**NAME**

(System Scheduler)term -- to system scheduler: terminate a process

**SYNOPSIS**

term = 9.

**INTRODUCTION**

The system scheduler is part of the basic kernel. It executes at processor priority 2 with kernel D-space enabled. The primary function of the scheduler is to schedule processes which execute at processor priority one and zero so as to maximize CPU usage without compromising real time response. The scheduler is primarily a message source, with almost all messages going to the memory manager, however, process termination requires manipulation of scheduler queues, hence the requirement that all terminate messages be sent to the scheduler.

**INPUT**

```
struct {
    struct msghdr hdr;
    struct cp_clist p_cdir;    /* 3-word capability structure */
    char p_uid;
    char p_gid;
    int p_prc;                /* process number to dump */
    int p_sid;                /* segment id of process PCB (supervisor only)*/
    int p_tpath;             /*offset to start of pathname */
    int p_mstblk[];
};
```

**VALUE**

The same message is forwarded to either the memory manager or the process manager.

**DESCRIPTION**

The user count on the process *p\_prc* is decremented and if zero, it is removed from the queue of active processes. All outstanding messages to the terminating process are returned with the system status byte equal to 0200. If the process is a kernel process or if *p\_tpath* is nonzero, the message is forwarded to the process manager, otherwise it is forwarded to the memory manager.

**DIAGNOSTICS**

The user status byte is set to -1 if *p\_prc* is invalid.

(SYSTEM-SCHEDULER)TERM(c)

(SYSTEM-SCHEDULER)TERM(c)

NAME

(System Scheduler) - to system scheduler; to manage a process

SYNOPSIS

term = X

DESCRIPTION

The system scheduler is part of the basic kernel. It executes as processor priority 2 with kernel D space enabled. The primary function of the scheduler is to schedule processes which execute at processor priority one and zero so as to maximize CPU usage without compromising real time response. The scheduler is primarily a message router, with almost all messages going to the process manager, however, process termination requests (notification of scheduler queues) since the requirement that all terminate messages be sent to the scheduler.

INPUT

start

start message data  
struct task\_p entry  
char p\_ptr  
char p\_ptr  
int p\_ptr  
int p\_ptr  
int p\_ptr  
int p\_ptr

\* 3-word capability structure \*  
\* process number to dump \*  
\* argument is of process PCB (supervisor only) \*  
\* offset to word of parameter \*

This page has been left blank intentionally.

VALUE

The same message is forwarded to either the process manager or the process manager

DESCRIPTION

The user count on the process p\_ptr is decremented and if zero, it is removed from the queue of active processes. All outstanding messages to the terminating process are returned with the system status bits equal to 0200. If the process is a kernel process or if it is a normal process, the message is forwarded to the process manager; otherwise it is forwarded to the process manager.

DIAGNOSTICS

The user status bits are set to 1 if p\_ptr is invalid.