NAME

sprofil — turn on/off system profiling

SYNOPSIS

#include <sys/sprof.h>

int sprofil (spcnt, numcnts, lowpc, intsize)
struct SPCNT spcnt;
unsigned int numcnts;
caddr_t lowpc;
unsigned int intsize;

DESCRIPTION

Calling *sprofil* with *spcnt* non-zero will initiate system profiling. If any other process is profiling, **EBUSY** is immediately returned. If *intsize* is 0, then the system will profile system routines, reserving a counter for *numcnts* global external text symbols. The (sorted) starting addresses for the system routines are provided by the user (usually from **/unix**).

If *intsize* is non-zero, the system will reserve a counter for every *intsize* group of bytes, starting at byte address *lowpc*, for a total of *numcnts* intervals.

If the size of the *spcnt* structure would overflow one PDP11/70 memory page (8192 bytes), then **EINVAL** is returned. Otherwise, the user's data space is locked in memory and the memory management information for the *spcnt* structure is saved in the kernel's *sysprof* structure.

If an independent clock is used (either a DEC KW11-K or a Digital Pathways TCU100 may be used), then that clock is started. When it interrupts (should be at level 7), or when the system clock routine is called, if no independent clock is used, the counter for the interrupted routine is incremented by 1. If the system was in user or idle mode, that is recorded instead.

The system increments the proper counter in user D space by temporarily changing kernel D space register 5 to point to the user page with the table of counters (hence the one page limit for the size of the *SPCNT* structure).

System profiling is stopped by sending a 0 in argument one. Normally, a user would do:

```
sprofil (spcnt, numcnts, lowpc, intsize);
sleep (seconds);
sprofil (0, 0, 0, 0);
```

and then report the results in some tabular form (see *sprof*(1M)).

The file **<sys/sprof.h>** including the prototype *SPCNT* structure, is as follows:

```
/*              @(#)sprof.h     3.2             */

/*
 *
 * Used by system profiling routines( sprofil.sincupc and sprof )
 *
 *
 */

#ifdef KERNEL
struct  pgreg {
                char *par;
                char *pdr;
};

struct sysprof {
```

```
                struct          SPCNT           *base;
                caddr_t lowpc;                  /* low pc word for i option */
                unsigned int numcnts;           /* number of counters in union array */
                unsigned int intsize;           /* size of i intervals or 0 for r opt */
                int pid;
                struct pgreg  newpg;
                struct pgreg  oldpg;
};

#endif
struct          NHIT {
                caddr_t         nloc;
                spcnt_t         nhits;
};

struct          SPCNT {
                long            b_urhits;
                long            b_syhits;
                long            b_idhits;
                union           {
                                /*
                                 *              "allocate" maximum possible size of counter buffers
                                 *              (they must fit entirely into one page)
                                 */
                                struct          NHIT            ropt[(8192 - 3*sizeof(long))/sizeof(struct NHIT)];
                                spcnt_t                         iopt[(8192 - 3*sizeof(long))/sizeof(spcnt_t)];
                }               u_ct;
};

#ifdef          IPROFCLK

/* independent profile clock kw11-k (A clock) */

#define         KW11K           (struct kw11ka *)0170404

struct          kw11ka {
                int             kw11ks;
                int             kw11kb;
};
#else
#ifdef          IPROFCLB

/* independent profile clock TCU-100 (battery clock) */

#define TCU100 (int *)0160774
#define TCURATE -48
                /*              rate of -33 should be 62.06/sec, is 120/sec for our clock
                                                -45             45.6    70.6
                                                -64             31      42.6
                                                -48             42.6    64
                                our clock may be dumb, but at least it's consistent
                 */

#endif
#endif
```

Notice that only the kernel gets the *sysprof* definition; the user can use the *SPCNT* structure definition.

SEE ALSO

sprof(1)

WARNINGS

If the data space in the kernel gets too big, the kernel D-space register 5 trick may not work.

If the system clock is used, any system routine in sync with the clock may appear invisible to system profiling.

ASSEMBLER

(syscb = 45. ; sprofil = 4.)
(struct spcnt in r0; a 4 in r1)
sys sprofil; numcnts; lowpc; intsize;