NAME

acct – per-process accounting file format

SYNOPSIS

#include <sys/types.h>
#include <sys/acct.h>

DESCRIPTION

Files produced as a result of calling **acct**(2) have records in the form defined by **<sys/acct.h>**, whose contents are:

```
typedef ushort  comp_t;        /* pseudo "floating point" representation */
                               /* 3 bit base-8 exponent in the high */
                               /* order bits, and a 13-bit fraction */
                               /* in the low order bits. */

struct   acct
{
         char     ac_flag;     /* Accounting flag */
         char     ac_stat;     /* Exit status */
         uid_t    ac_uid;      /* Accounting user ID */
         gid_t    ac_gid;      /* Accounting group ID */
         dev_t    ac_tty;      /* control tty */
         time_t   ac_btime;    /* Beginning time */
         comp_t   ac_utime;    /* accounting user time in clock */
                               /* ticks */
         comp_t   ac_stime;    /* accounting system time in clock */
                               /* ticks */
         comp_t   ac_etime;    /* accounting total elapsed time in clock */
                               /* ticks */
         comp_t   ac_mem;      /* memory usage in clicks (pages) */
         comp_t   ac_io;       /* chars transferred by read/write */
         comp_t   ac_rw;       /* number of block reads/writes */
         char     ac_comm[8];  /* command name */
};

/*
 * Accounting Flags
 */
#define  AFORK   01            /* has executed fork, but no exec */
#define  ASU     02            /* used super-user privileges */
#define  ACCTF   0300          /* record type */
#define  AEXPND  040           /* Expanded Record Type – default */
```

In **ac_flag**, the **AFORK** flag is turned on by each **fork** and turned off by an **exec**. The **ac_comm** field is inherited from the parent process and is reset by any **exec**. Each time the system charges the process with a clock tick, it also adds to **ac_mem** the current process size, computed as follows:

(data size) + (text size) / (number of in-core processes using text)

The value of **ac_mem / (ac_stime + ac_utime)** can be viewed as an approximation to the mean process size, as modified by text sharing.

The structure **tacct**, (which resides with the source files of the accounting commands), represents a summary of accounting statistics for the user id **ta_uid**. This structure is used by the accounting commands to report statistics based on user id.

```
/*
 * total accounting (for acct period), also for day
 */

struct  tacct {
        uid_t           ta_uid;         /* user id */
        char            ta_name[8];     /* login name */
        float           ta_cpu[2];      /* cum. cpu time in minutes, */
                                        /* p/np (prime/non-prime time) */
        float           ta_kcore[2];    /* cum. kcore-minutes, p/np */
        float           ta_con[2];      /* cum. connect time in minutes, */
                                        /* p/np */
        float           ta_du;          /* cum. disk usage (blocks)*/
        long            ta_pc;          /* count of processes */
        unsigned short  ta_sc;          /* count of login sessions */
        unsigned short  ta_dc;          /* count of disk samples */
        unsigned short  ta_fee;         /* fee for special services */
};
```

ta_cpu, **ta_kcore**, and **ta_con** contain usage information pertaining to prime time and non-prime time hours. The first element in each array represents the time the resource was used during prime time hours. The second element in each array represents the time the resource was used during non-prime time hours. Prime time and non-prime time hours may be set in the **holidays** file (see **holidays**(4)).

ta_kcore is a cumulative measure of the amount of memory used over the accounting period by processes owned by the user with uid **ta_uid**. The amount shown represents kilobyte segments of memory used, per minute.

ta_con represents the amount of time the user was logged in to the system.

FILES

/etc/acct/holidays	prime/non-prime time table

SEE ALSO

acctcom(1), **acct**(1M), **acctcon**(1M), **acctmerg**(1M), **acctprc**(1M), **acctsh**(1M), **prtacct**(1M), **runacct**(1M), **shutacct**(1M), **acct**(2), **exec**(2), **fork**(2)

NOTES

The **ac_mem** value for a short-lived command gives little information about the actual size of the command, because **ac_mem** may be incremented while a different command (for example, the shell) is being executed by the process.