

PROCESS ACCOUNTING

- Process accounting : when enabled kernel writes an accounting record each time a process terminates
- Accounting records : 32 bytes of binary data

Struct acct

```
{  
    char ac_flag;  
    char ac_stat;  
    uid_t ac_uid;  
    gid_t ac_gid;  
    dev_t ac_tty;  
    time_t ac_btime;  
    comp_t ac_utime;  
    comp_t ac_stime;  
    comp_t ac_etime;  
    comp_t ac_mem;  
    comp_t ac_io;  
    comp_t ac_rw;  
    char ac_comm;  
}
```

```

/*prog: to generate accounting data */

#include<sys/types.h>
#include<sys/acct.h>
#include "ourhdr.h"
#define ACCTFILE  "/var/adm/pacct"
static unsigned long  compt2ulong(comp_t);
int main(void)
{
    struct acct        acdata;
    FILE               *fp;
    if ( (fp = fopen(ACCTFILE, "r")) == NULL)
        err_sys("can't open %s", ACCTFILE);

    while
    (fread(&acdata, sizeof(acdata), 1, fp) == 1)
    {
        printf("%0-*.*s e = %6ld, chars = %7ld, "
            "stat = %3u: %c %c %c %c\n",
            sizeof(acdata.ac_comm),
            sizeof(acdata.ac_comm),
            acdata.ac_comm,
            compt2ulong(acdata.ac_etime),
            compt2ulong(acdata.ac_io),
            (unsigned char) acdata.ac_stat,

#ifdef ACORE
            /* SVR4 doesn't define ACORE */
            acdata.ac_flag & ACORE ? 'D' : '',
#else
            '',

```

```

#endif
#ifdef AXSIG
    /* SVR4 doesn't define AXSIG */
    acdata.ac_flag & AXSIG ? 'X' : '',
#else
    ',
#endif
acdata.ac_flag & AFORK ? 'F' : '',
    acdata.ac_flag & ASU ? 'S' : '');
    }
if (ferror(fp))
    err_sys("read error");
    exit(0);
}
static unsigned long
    compt2ulong(comp_t comptime)
/* convert comp_t to unsigned long */
{
    unsigned long val;
    int exp;
    val = comptime & 017777;
        /* 13-bit fraction */
    exp = (comptime >> 13) & 7;
        /* 3-bit exponent (0-7) */
    while (exp-- > 0)
        val *= 8;
    return(val);
}

```