

```
/*
    Fork Join Example
    Demonstrates how to spawn processes and exchange messages
*/

/* defines and prototypes for the PVM library */
#include <pvm3.h>

/* Maximum number of children this program will spawn */
#define MAXNCHILD    20
/* Tag to use for the joining message */
#define JOINTAG      11
int
main(int argc, char* argv[])
{

    /* number of tasks to spawn, use 3 as the default */
    int ntask = 3;
    /* return code from pvm calls */
    int info;
    /* my task id */
    int mytid;
    /* my parents task id */
```

```
int myparent;
/* children task id array */
int child[MAXNCHILD];
int i, mydata, buf, len, tag, tid;
/* find out my task id number */
mytid = pvm_mytid();

/* check for error */
if (mytid < 0) {
    /* print out the error */
    pvm_perror(argv[0]);
    /* exit the program */
    return -1;
}
/* find my parent's task id number */
myparent = pvm_parent();

/* exit if there is some error other than PvmNoParent */
if ((myparent < 0) && (myparent != PvmNoParent)) {
    pvm_perror(argv[0]);
    pvm_exit();
    return -1;
}
```

```

/* if i don't have a parent then i am the parent */
if (myparent == PvmNoParent) {
    /* find out how many tasks to spawn */
    if (argc == 2) ntask = atoi(argv[1]);
    /* make sure ntask is legal */
    if ((ntask < 1) || (ntask > MAXNCHILD)) { pvm_exit();
return 0; }

    /* spawn the child tasks */
    info = pvm_spawn(argv[0], (char**)0, PvmTaskDefault,
(char*)0,
        ntask, child);
    /* print out the task ids */
    for (i = 0; i < ntask; i++)
        if (child[i] < 0) /* print the error code in decimal*/
            printf(" %d", child[i]);
        else /* print the task id in hex */
            printf("t%x\t", child[i]);
    putchar('\n');
    /* make sure spawn succeeded */
    if (info == 0) { pvm_exit(); return -1; }

    /* only expect responses from those spawned correctly */

```

```

ntask = info;

for (i = 0; i < ntask; i++) {
    /* recv a message from any child process */
    buf = pvm_recv(-1, JOINTAG);
    if (buf < 0) pvm_perror("calling recv");
    info = pvm_bufinfo(buf, &len, &tag, &tid);
    if (info < 0) pvm_perror("calling pvm_bufinfo");
    info = pvm_upkint(&mydata, 1, 1);
    if (info < 0) pvm_perror("calling pvm_upkint");
    if (mydata != tid) printf("This should not happen!\n");
    printf("Length %d, Tag %d, Tid t%x\n", len, tag, tid);
}
pvm_exit();
return 0;
}
/* i'm a child */
info = pvm_initsend(PvmDataDefault);
if (info < 0) {
    pvm_perror("calling pvm_initsend"); pvm_exit(); return -1;
}
info = pvm_pkint(&mytid, 1, 1);
if (info < 0) {

```

```
        pvm_perror("calling pvm_pkint"); pvm_exit(); return -1;
    }
    info = pvm_send(myparent, JOINTAG);
    if (info < 0) {
        pvm_perror("calling pvm_send"); pvm_exit(); return -1;
    }
    pvm_exit();
    return 0;
}
```