

ort Program 2: Peterson's algorithm

Due: Friday, Feb 17 at 11:59 pm

Overview

In this assignment you are to write two versions of Peterson's algorithm. The first version must be submitted in module `pete2.c`. The second version must be submitted in module `pete8.c`. Both modules must contain two functions:

```
void pete_lock( int tid)          void pete_unlock(int tid)
{                                  {
}                                  }
```

Use the data structures

```
typedef struct pete_lock_type
{
    int using[2];    /* 0 = left 1 = right */
    int turn;        /* Ditto          */
} pete_lock_t;
```

```
typedef struct pete_path_type
{
    int lockid;
    int side;    /* = left 1 = right */
} pete_path_t;
```

Use `pthread2.c` in `assns/sp2` as the main program for testing. Begin with the two thread version. To do this version you will need only one instance of a `pete_lock_t` which should be declared

```
static pete_lock_t lock;
```

Begin testing on machine *hydra*. On this machine you should not need westall's hack 1. **AFTER** your program works on *hydra* try it on *jmw*. Here you will need to add Westall's hack 1.

The 8 thread version

When you have completed `pete2.c`. Move on to `pete8.c`. Change `NUM_THREADS` in `pthread2.c` from 2 to 8. For this you should turn `pete_lock()` and `pete_unlock()` into *wrapper* functions which pass a second parameter, the current level in the lock tree illustrated in the notes as a parameter.

I recommend that `do_pete_lock()` be recursive and that `do_pete_unlock()` just be a simple iteration.

```
void pete_lock( int tid)           void pete_unlock(int tid)
{
    do_pete_lock(tid, 0);         do_pete_unlock(tid, 2);
}
```

How to submit your program:

**NOTE: This procedure has NOTHING in common with "handin" nor "sendlab"
Do NOT even TRY to think about how they fit into this procedure because
THEY DON'T!!**

<<<Do NOT turn in any image files, core files, makefiles etc.>>>

You must turn in 2 files: `pete2.c` and `pete8.c`

1. From any departmental Solaris system `ssh` to workstation `jmw`
2. The submission directories lie in the directory `/local/jmw2/322/sp2` which is available ONLY IF YOU HAVE LOGGED INTO WORKSTATION `jmw`. Each student has a subdirectory of `/local/jmw2/322/sp2`. The name of your subdirectory is your userid (in the example we will assume your id is `wjsmith`).
3. copy (via the `cp` command) required file to your subdirectory in `/local/jmw2/322/sp1`

For example:

```
cp /home/wjsmith/322/sp1/pete*.c /local/jmw2/322/sp2/wjsmith
```

4. Don't modify the permissions on your subdirectory. They are set so that ONLY you can access your directory.

After you think you have turned your programs in, its a good idea to
`cd /local/jmw2/322/sp2/wjsmith`
and make sure your files are there and they still compile and work correctly.