Problem Set 8

CS 311

Due at the beginning of class the following Monday in hardcopy.
Sections 4.4–6

1. Briefly discuss the relative advantages of Pthreads and OpenMP.

2. What is the main difficulty with thread cancellation? How does Pthreads address this difficulty?

3. The program `thrd1.c` on page 2 outputs

   ```
   CHILD: value = 5
   PARENT: value = 0
   ```

   while the program `thrd2.c` on page 3 outputs

   ```
   CHILD: value = 5
   PARENT: value = 5
   ```

   Explain why they have different outputs.

```c
/* thrd1.c */
#include <pthread.h>
#include <stdio.h>
#include <sys/types.h>

int value = 0;
void *runner(void *param); /* the thread */

int main(int argc, char *argv[])
{
   pid_t pid;
   pthread_t tid; /* the thread identifier */
   pthread_attr_t attr; /* set of attributes for the thread */

   pid = fork();
   if (pid == 0) /* child process */
   {
      /* get the default attributes */
      pthread_attr_init(&attr);
      /* create the thread */
      pthread_create(&tid,&attr, runner, NULL);
      /* now wait for the thread to exit */
      pthread_join(tid,NULL);
      printf("CHILD: value = %d\n",value);
   }
   else if (pid > 0) /* parent process */
   {
      /* wait for child to exit */
      wait(NULL);
      printf("PARENT: value = %d\n",value);
   }
   return 0;
}

/* The thread will begin control in this function */
void *runner(void *param)
{
   value = 5;
   pthread_exit(0);
}
```

```c
/* thrd2.c */
#include <pthread.h>
#include <stdio.h>
#include <sys/types.h>

int value = 0;
void *runner(void *param); /* the thread */

int main(int argc, char *argv[])
{
   pthread_t tid; /* the thread identifier */
   pthread_attr_t attr; /* set of attributes for the thread */

   /* get the default attributes */
   pthread_attr_init(&attr);
   /* create the thread */
   pthread_create(&tid,&attr, runner, NULL);
   /* now wait for the thread to exit */
   pthread_join(tid,NULL);
   printf("PARENT: value = %d\n",value);
   return 0;
}


/* The thread will begin control in this function */
void *runner(void *param)
{
   value = 5;
   printf("CHILD: value = %d\n",value);
   pthread_exit(0);
}
```