9.3 POSIX Threads Case Study

- *Study Program 9.9 (pp. 351-2)*

- for copying multiple files.

  * Consider the executable named "copy"

  - and invocation Q : "copy a b 3"

- *1. Write Unix commands to mimic its effect.*

  * Hint: Use cp, wc

  * How many processes are used by Unix commands?

- *2. How many processes and threads are created by "Q"?*

- *3. How many files are opened?*

  * Draw the File Descriptor Table

- *4. Identify FDT entries used by each thread.*

  * Is any FDT entry used by multiple threads?

  * Is any global variable used by multiple threads?

- *5. What will it print?*

- *6. Is malloc() thread-safe?*

  * Note malloc() calls from multiple threads.

Thread Library - Error Conventions

- *Pthread library system calls*
  - * replace global variable "errorno" by a macro
  - * mimic thread-private variable "errorno"
- *Pthread library system calls*
  - * return 0 on success
  - * or errornumber otherwise
  - * Not required to set "errno" global variable
  - - which is used by many other POSIX system calls
- *Example code to check error*

```
#include <errno.h>
#include <stdio.h>
...
if (rtn = pthread_create( ... )) {
        /* error has occurred */
        perror("Error: pthread_create, ");
        switch (rtn) {
                case EAGAIN: perror("Insufficient resources"); break;
                case EINVAL: perror("Invalid arguments"); break;
        }
        exit(-1);
} else { /* non error */ }
```

Mid-quarter Exam.

- *Basic Information*
  - * Place: Classroom
  - * Time: Mon 11/1/99 (eve. Sec.), Tues. 11/2/99 (day sec.)
  - * Open book, man-pages, classnotes
  - * Closed neighbors, computers etc.
  - * Syllabus: Chapters 1, 2, 3, 5 (Robbins/Robbins)
- *Nature*
  - * Problem solving - calculate file sizes, etc.
  - * Analysis - output for a given program
  - * True/False
  - * Match items from two tables
  - * Few definitions, comparisons, discussions
- *Practice Exam. this Wednesday in recitation*

Mid-quarter Exam. Details

- *Details - Important System Calls*

  * Ch. 1.: perror, strerror

  * Ch. 2.: getpid, getppid, getenv, setsid
  - fork, exit, wait, waitpid, execl

  * Ch. 3.: getcwd, chdir, opendir, readdir, closedir
  - status, open, read, write, close, dup2, pipe

  * Ch. 5.: kill, raise, alarm,
  - sigprocmask, sigaction, pause, sigsuspend
  - (sigempty, sigfillset, sigaddset, sigdelset, sigismember)

Mid-quarter Exam. Details

- *Details - calls from ANSI C Standard Libraries*
  - C Memory Management: malloc, free
- *Details - Important Shell symbols and Commands*
  - Ch. 1.: man, make, cc,
  - Ch. 2.: ps, env, &, bg, fg,
  - Ch. 3.: cd, ls, find, ln, |, <, >, >>,
  - Ch. 5.: kill, intr (^C)