Operating Systems: Practice: Lesson 3

Sevak Amirkhanian

What is errno?

errno is defined by the ISO C standard to be a modifiable lvalue of type *int*, and must not be explicitly declared; *errno* may be a macro. *errno* is thread-local; setting it in one thread does not affect its value in any other thread.

How is errno changed and how can we use it?

errno is always set by the last system call. It is the error code of this last system call if any error occurred. To access errno-associated message we use perror(..):

```
void perror(const char *s);
```

Sync primitives can be shared.

Synchronization primitives in POSIX like mutexes, semaphores and conditional variables can be shared across different processes.

How to make POSIX mutex shared?

Whether mutex is shared or not, should be determined with the mutex attribute. In order to make mutex shareable, we need to use the following POSIX interface:

```
int pthread_mutexattr_setpshared(
pthread_mutexattr_t *attr,
int pshared);
```

`PTHREAD_PROCESS_SHARED` should be passed as pshared value if we want the mutex to be accessible from other processes.

Homework 3: Multiprocess sum calculation

Calculate the sum using multiple processes by forking the parent process.

The detailed description can be seen in: description.txt

The homework folder is called homework_3 in the zip, where you can see all the necessary files.

Thank you.