## Threads Mini-Lab

### Introduction:

The purpose of this mini-lab is to learn what threads share with other threads. Also, what are the advantages and disadvantages of having multiple threads share an object?

### Method:

Run the ParentLaugh program. Each ‘External’ thread generates a new object with a thread attached. ‘Internal’ threads generate a thread on the current (same) object. Three threads are created which laugh different laughs: “Ha”, “Ho”, “He”. The laughs can be spaced out with an n-second delay before every print, or no delay. In order to learn what threads share, we will observe: 1) which data variables are shared; 2) whether signals (such as CTL-C are shared; 3) the number of processes running. Use 2 commands:

$ ps –al $ ps –al H

### Results:

Below are the results for internal and external mode, with varying sleep times:

(Note laughs per line, interspersing of Ha/Ho/He, any other differences you see.)

|  |  |  |
| --- | --- | --- |
| ParentLaugh Results | One Object (Shared) | Multiple Objects (External) |
| Sleep 0 |  |  |
| Sleep 1 |  |  |
| CTL-C Effect on Threads |  |  |
| Number of processes visible via $ ps –al  And $ ps –al H |  |  |

### Analysis:

Now explain what you think is going on in this code and O.S. between the differences you see (shared versus external), and how threads differ from working with processes. Consider how many objects are created and how sharing affects data variables when multiple threads share an object.

### Conclusion:

What did you learn from all this?