|  |  |
| --- | --- |
| CS 241 | Lecture #12  Threads! |

pthread\_create

pthread\_join

pthread\_exit

**#include** **<pthread.h>**

**int** **pthread\_create(pthread\_t** **\***thread**,**

**pthread\_attr\_t** **\***attr**,**

**void** **\*(\***start\_routine**)** **(void** **\*),**

**void** **\***arg**);**

**int** **pthread\_join(pthread\_t** thread**,** **void** **\*\***retval**);**

**void** **pthread\_exit(void** **\***retval**);**

Compile and link with -pthread.

#1 My program calls pthread\_create twice. How many stacks does my process have?

#2 What is the difference between a process and a thread?

#3 What does pthread\_cancel do?

and are there alternatives?

#4 Differences between exit() and pthread\_exit()?

#5 ..so why would you call pthread\_exit in your main method?

#6 Give four ways that a thread can be terminated

#7 Why are some functions e.g. asctime,getenv, strtok, strerror not thread-safe?

|  |
| --- |
| 1. char\* to\_message(int num) { 2. char static result [256]; 3. if(num < 1000) 4. sprintf(result, "%d : blah blah" , num); 5. else strcpy(result, "Unknown"); 6. return result; 7. } |

#8. What are condition variables, semaphores, mutexes?

#9 Advantages of threads over forking processes?

#10. Can you fork a process with multiple threads?

#11. Examples of why you might fork processes instead of using threads