

## params, threads, messages, memory & signals

Development: Machine to develop code, contains a virtual debug machine.

Debug: A virtual machine that is used to debug kernel mode code.

1. Clone your bitbucket repo to the development machine
2. Checkout a new branch named LAB3. Add dir lab3/.
3. Download all of the \*.h & \*.c files from Canvas / Files / Lab3 into lab3.
4. Create a makefile to with the following targets:
5. Demo, Add, commit, checkout master. Merge branch LAB3 - Re-checkout

---

6. Build a parser to accept the following commands and options:
7. -threadcount N, -c N, -timer L H, -t L H, -message "M", -m "M"
8. Spawn N-1 Threads. Sleep for L to H time. (Each)
9. Print the id on each thread, using the sleep count for the #.
10. Demo, Add, commit, checkout master. Merge branch LAB3 - Re-checkout

---

11. Construct a signaler in the parent and handler in the children.
12. Allocate a shared memory region and store the message "M" there.
13. Each child reads the message and print it as before.
14. Demo, Add, commit, checkout master. Merge branch LAB3 - Re-checkout

---

15. After constructing each child, send the SIGSTOP to each.
16. In a loop, for each child: Sleep for L to H ms - in the parent.
17. Send the SIGCONT to the child process after sleeping.
18. Wait for all children to exit.
19. Demo, Add, commit, checkout master. Merge branch LAB3 - Re-checkout

---

20. Create a thread safe version of messenger.
21. Create a thread safe version of signaler.
22. Demo, Add, commit, checkout master. Merge branch LAB3.