Dynamic Diagnostic Tools (on Unix)

- Process control
- Process activity
- Shared resources

Process Control

- `ps` lists (some) processes running on a machine
- To list your processes:
  - Solaris, System V: `ps -u username`
  - Solaris, BSD-style: `/usr/ucb/ps gx`
  - Linux: `ps gx`
- To terminate a process:
  - `kill pid`
- To terminate a process group:
  - `ps -ju your_username` to list process group ID's (pgid's)
  - `kill -pgid` (note the negative progress group ID)

Process Activity

- `top` shows useful information about each process
  - state (running, waiting for CPU, sleeping, etc.)
  - memory usage (total and resident)
  - CPU usage, both instantaneous (percentage) and cumulative

load averages: 1.02, 0.98, 0.81 06:55:37
70 processes: 66 sleeping, 2 zombie, 2 on cpu
Memory: 1024M real, 15M free, 43M swap in use, 2007M swap
Process Activity, (cont’d)

- **truss** shows a processes’s system calls (**strace** on Linux; **trace** on some older systems)
- memory access checkers, *e.g.*, Rational’s Purify, NuMega’s BoundsChecker, Parasoft’s Insure++
- debuggers
- printouts, especially if well thought out, *e.g.*,
  
  ```c
  ACE_DEBUG ((LM_INFO, "Foo::f () ; i = %d\n", i));
  ```
  
  - can disable with a compile-time switch
  - can set a debugging level
  - can output all built-in data types

Shared Resources

- **ipcs** lists shared memory segments, semaphores, and message queues
- **netstat** shows network connection and interface status
  - `netstat -f inet` lists TCP/IP sockets
    (netstat - -inet on Linux)
  - `netstat -i` lists network interfaces used for TCP/IP
  - On Linux only, `-p` option shows which process owns each socket
    (for your own sockets, unless superuser)

For Further Information

- man pages
- many good textbooks, such as W. Richard Steven’s *Advanced Programming in the UNIX® Environment*