CS252: Systems Programming

Ninghui Li

Topic 21: Lab 6: The Process Games and Inter-Process Communications

The Process Games

- In each process game, between 2 and 5 programs can participate.
- The arena program creates (i.e., forks) N processes of each program, where N is at least 10, and N*P is at most 100.
  - Each program is loaded with N and P given as arguments.
- The total number of processes in the arena is limited to N*P once the arena starts.

The Players

- The goal of a participating program is to make as many processes in the arena to be duplicates of itself.
- A player must thus
  - Find target
    - Use kvm_getprocs(), guess PID, etc.
  - Kill target (kill)
  - Duplicate itself (fork)

Two Arenas: Arena 1

Start each program under its own name.

- Run **`sudo su –u arena –c “./arena N ./your_prog ./fork ./n_hit”`**

- Grading done by ps

Script run1.sh: Part 1

```
#!/usr/local/bin/bash
if [ "$#" -lt 2 ]; then
  echo "USAGE:"
  echo "$0 num_processes prog1 prog2 [prog3 ... prog5] "
  echo "Will output score for prog1"
  exit 1
fi
NUM_PROCESS=$1
PRG_NAME=$2
TOTAL=$((($#-1)*$NUM_PROCESS))
sudo su arena -c "./arena $*"
```

Script run1.sh: Part 2

```
sleep 10
ps -U arena > match_result.txt
sudo pkill -KILL -U arena
X=$(grep $PRG_NAME match_result.txt | wc -l)
if [ $X -gt $(($TOTAL)) ]; then
  X=$TOTAL
fi
MATCH_SCORE=$(echo "scale=2; $X*100/($TOTAL)" | bc -l)

echo "GAME: $2 with maximum number of processes $TOTAL"
echo "-------------------------"
echo $MATCH_SCORE
```

Arena 2

Run each contestant program under a randomly generated program name

Two steps

- Run **`arena_s 20 ./your_prog ./fork`** first to set up the arena (making copies of these two programs under directory tmp)
- Then run
  - **`sudo su –u arena –c “./arena_r 40”`**
  - About 10 seconds into the game, a signal SIGUSR1 will be delivered

Friend or Foe?

- Possible strategies
  - Use **kvm_getargv**() to get pid
  - Use command line arguments
  - Use IPC between friends
    - E.g, use shared memory as a bulletin board shared by all friends
    - E.g., use signals
    - Possibly others
  - Don’t care….

Strategic Thinking

- Fast and furious versus methodical
  - Is the overhead worthwhile?
- Whether to counteract particular strategies?
- Should one use different strategies based on values of N, P?
- Should one have different instances of the players adopt different strategies?

Other Things to Explore

- Understand what existing players do?

  Use strace to find out the system calls made by a player, and experiment to understand relative performances

- Understand process scheduling and system calls made
  - Possibly use auditctl

How to Set up?

- Install Oracle Virtual Box
- Download the VM image
- Log in as cs252
- Commands

  -
    ```
    cd lab6
    ```

  -
    ```
    sudo su arena –c “run1.sh  10 ./n_hit ./fork”
    ```
  - From a different terminal, top –U arena
  - pgrep –lu arena1 prog   show remaining processes
  - sudo pkill -KILL -U arena

Submit

- Use turnin to submit

Lab 6 as a Small Research Project

- When you start, you don’t know what the result is going to be.
- Need to try things and see if they work.
- The solution depends on assumptions.
- The solution may be simple.
- May need to spend more time thinking than coding.
- There is always a bit of luck involved.

Inter-Process Communications

Use external slides