BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI, K. K. BIRLA GOA CAMPUS, I SEMESTER 2020-2021

Operating Systems (CS F372)

Component: Online #3 Weightage: 10% [30 Marks]

Date: 28/10/2020, Time: 8:00 P.M. – 11:59 P.M.

Problem: Simulate IPL 2020 FINAL

Note: What to submit?

<Your IDNo>_OSOnline3.tar.gz file containing the following:

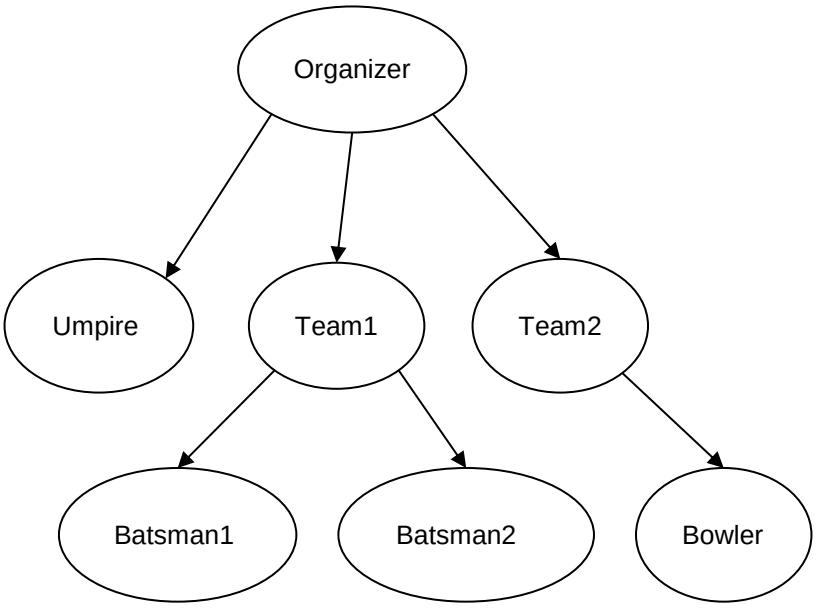
- Files – driver.c, cricket.h, cricket.c, makefile and README.

EndNote:

Problem Statement:

You are asked to simulate a cricket match. The cricket match contains 2 teams, an umpire & an organizer. Each team will have 11 players, details of which will be given as input.

This is a very simple simulation of the cricket match. So you don't need to consider any type of intricacies like byes, leg byes, wide, no ball, free-hit, etc. Assume each ball to be a legal delivery, the result of which is either runs (0-6) or a wicket.



Process Tree: When Team1 is Batting.

Organizer

The organizer would be the main/driver function that starts the program execution and creates 3 processes (2 teams & 1 umpire). After the match ends, the organizer outputs the stats for each team and writes in the required files. It then declares the Most Valuable Players (MVPs) [explained later] for each team and adds them to the commentary file.

Umpire

The umpire will read its configuration file (umpire.txt) using file I/O. The 1st line of the file contains umpire seed value and the 2nd line contains the Maximum number of overs in each innings.

After both the teams are ready (i.e. team processes have been created and the inputs have been read), the umpire tosses a coin based on the seed value (refer seeds section in Page 5). It then assigns sides based on the result of the toss (1->Batting, 0->Bowling for Team1). This information will be sent to the respective teams (using shared memory).

The umpire waits till the teams have created the required players (2 Batsmen by the Batting Team & 1 Bowler by the Bowling Team). After all the players are ready, the umpire sends the "NEXT BALL" signal to the current bowler. Umpire receives the "SHOT HIT" signal from the striker and generates a random outcome for that ball (0 runs, 1 run, 2 runs, 3 runs, 4 runs, 5 runs, 6 runs, or a Wicket), and raises the corresponding "BALL RESULT" signal. This cycle keeps on repeating till the end of the innings. After every 6 balls (1 over), the umpire sends a "NEXT OVER" signal to both the teams.

The umpire sends an "INNINGS OVER" signal to both teams if any of the following situations occur:

- All 10 wickets have fallen
- All overs have been bowled
- The batting team successfully chases the target set by the opponent team (possible only in the 2nd innings)

The umpire also maintains a scoreboard (runs, wickets & balls) for each innings and updates this information after every ball. After the first innings ends, the umpire switches the batting/bowling sides and starts the second innings (which proceeds the same way as the first one). After the second innings ends, the umpire process exits.

Teams

Both the teams should read their team order given in the configuration file (Team1.txt and Team2.txt). The 1st line of each team input file contains the team name (a single string), 2nd line contains the team seed value, 3rd – 13th lines contain the names of the players in the team (all single strings) and the 14th line contains the seed value for each player (separated by space).

Teams wait till the toss finishes, for the innings to start, after which, they create the required current players (striker, non-striker and bowler) according to the side (Batting/Bowling) allotted to them.

Batting order is the same as the team order in the configuration file of that team. The bowlers will be randomly selected from among the last 6 players in the team. After the teams receive the "INNINGS OVER" signal, they kill all its active players (All child processes of Team1 & Team2 are killed). Team processes (Team1 and Team2) will exit only after completion of the match.

Batting Team

The batting team initially creates two processes - each representing a batsman (a striker and a non-striker). If a batsman gets OUT (killed), the batting team must send the next batsman in the given batting order. If the team gets the "NEXT OVER" signal, the team should switch the striker and the non-striker.

Batsman

The batsman is an infinitely running process which is killed only if it gets OUT or the innings is finished. The striker, after getting a "BALL BOWLED" signal from the bowler, raises a "SHOT HIT" signal to the umpire. It then updates its individual score according to the response it gets from the umpire (via "BALL RESULT" signal). If the runs scored in the ball are odd (1,3,5) the strike switches. If the batsman gets OUT it kills itself and the new batsman (if generated) will get the strike. [If the batsman scores odd runs on the last ball of an over, the strike will change twice - so effectively no strike change].

Bowling Team

The bowling team initially creates only one process - representing a bowler. After receiving the "NEXT OVER" signal, the bowling team must kill the current bowler & send the next bowler with the restriction that a bowler cannot bowl two consecutive overs and the maximum number of overs allowed to a bowler = ceil(Maximum overs/5).

Bowler

The bowler is an infinitely running process which gets killed by the team only if the over ends or the innings is finished. After the bowler gets a "NEXT BALL" signal from the umpire to bowl, the bowler sends a "BALL BOWLED" signal to the striker. It then updates its individual score according to the response it gets from the umpire (via "BALL RESULT" signal).

For a better understanding of the signals, refer to the Signals section in Page 6

INPUT FORMAT/FILES	OUTPUT FORMAT/FILES
umpire.txt Seed value Maximum Number of Overs **team1.txt** Team1 Name Team1 Seed 11 Player Names 11 Player Seed values **team2.txt** Team2 Name Team2 Seed 11 Player Names 11 Player Seed values	Please refer to the output files for the output format (attached along) as they are self-explanatory. **OUTPUT FILES** commentary.txt teamOut1.txt teamOut2.txt **commentary.txt** contains the ball wise stats for the complete match. **teamOut1.txt** and **teamOut2.txt** contain the match stats for all the players of the team. Your output should ideally match with these but can also vary depending on the seed value implementation.

Organizer will save the individual player scores in teamOut1.txt and teamOut2.txt. It also needs to find the MVPs (most valuable bowler & batsman) in each team and append the output at the end of commentary.txt. Use exec, dup2, pipe and the teamOut1.txt, teamOut2.txt files to find this information. [Hint: sort (with n,r,k flags), cut, head commands can be used.]

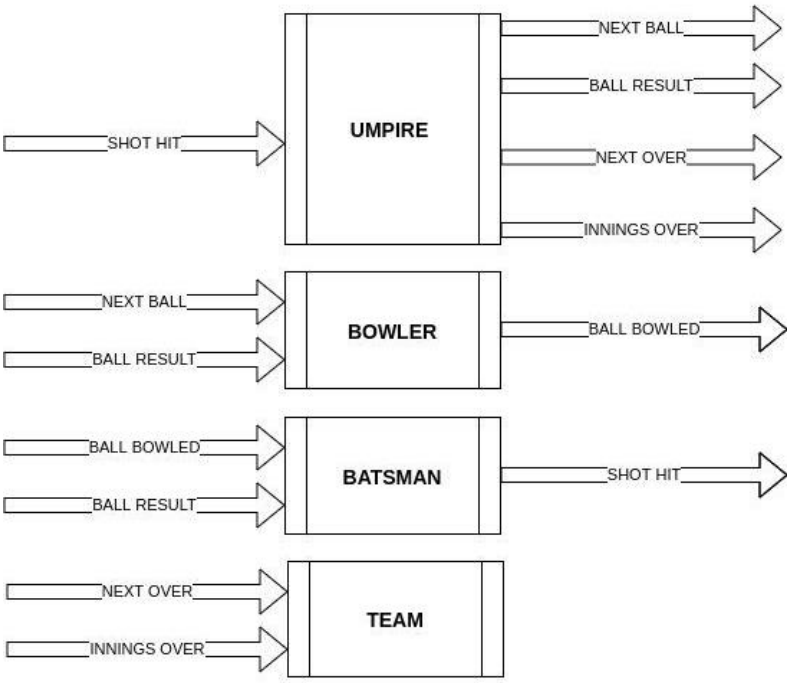
[Criteria for MVP selection: Most Valuable Batsman -> Max runs scored, Most Valuable Bowler -> Max wickets taken] (in case of a tie, just use the first value of the sorted output).

NOTES:

- Assume all the responsibilities (sending the next batsman/bowler) of a team captain are managed by the team process.
- Assume that only the batsman on strike (the striker) can get OUT.
- A process should wait for the previous tasks to finish (in other processes) wherever necessary to maintain consistency. For example:
    - Umpire process waits for a new batsman to be created in case of fall of wicket and allows the bowler to bowl the next ball only after the new batsman is ready.
    - Next ball can be bowled only when the strike change (if required) has happened for the previous ball.

- Seeds
    - Syntax for setting the seed value: srand(SEED)
    - Use the seed value (bowling team seed + overs finished in that innings) for deciding the next bowler. {random o/p from 0-5 corresponds to player# 6-11 respectively}
    - For run prediction: bowler seed + batsman seed + #balls bowled in that innings. {0-6 corresponds that many runs and 7 corresponds to wicket}
    - Use the umpire seed for the toss. {1 corresponds to batting of team1, 0 corresponds to bowling of team1}

SIGNALS



SIGNAL NAME	RAISED BY	HANDLED BY	DESCRIPTION
NEXT BALL	Umpire	Bowler	Signals the bowler to bowl the next ball
BALL BOWLED	Bowler	Batsman	The bowler raises a signal indicating that the ball has been bowled; the batsman must hit a shot
SHOT HIT	Batsman	Umpire	The batsman raises a signal indicating that the shot has been hit; the umpire must decide the result of the ball i.e. (0-6 runs or a wicket)
BALL RESULT (8 signals)	Umpire	Bowler/ Batsman	Signals the current striker and the bowler to update the score and to change the strike according to signal number raised
NEXT OVER	Umpire	Bowling Team/ Batting Team	Signals the teams that the current over has finished; strike change and change in bowler will be simulated
INNINGS OVER	Umpire	Bowling Team/ Batting Team	Signals the end of the current innings - start the 2nd innings if 1st innings ended, end the whole match if 2nd innings ended