

2.1 the tracker

- Always on tracker with fixed ip address and port
- Tracker maintains a database of players
- A tracker process must be started before any player processes
- Runs in an infinite loop, listening for a message on the port bound to the UDP socket sent to it from a player, handles the request, and responds back to the player
- At least two player processes must be started
- Each one reads commands from stdin until exit
- After registering with **tracker**, the **player** may start a game/be selected as a participant by another **player**
- **Tracker** must support messages corresponding to the following commands from a player
- **THIS IS WHAT MUST BE IMPLEMENTED:**
 - register <player> <IPv4> <t-port> <p-port>
 - Player is an alphabetic string of $L \leq 15$
 - IPv4 is a string giving an address in dotted decimal notation
 - Does not need to be unique
 - T-port used for communication between this player and the tracker
 - P-port for communication between players
 - —
 - Each player may only register once
 - Using map, use methods from site 1
 - RETURN:
This command returns SUCCESS if the player's name is not a duplicate among all peers registered. In this case, the tracker stores a tuple associated with the player in a database. Also sets the state of the player to FREE indicating availability to play a game
 - Otherwise, tracker takes no action and responds with a return code FAILURE indicating duplication registration or other issue
 - Not actual return, but use sendto function (found in main)
 - DOCUMENTATION USED:
 - <https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/>
 - <https://favtutor.com/blogs/split-string-cpp>
 - <https://cplusplus.com/reference/sstream/stringstream/>
 - Query players
 - Query players currently registered with tracker
 - Returns a return code equal to # of registered players
 - Returns a list of tuples associated with each player
 - If no players are registered, return code is set to zero and the list is empty
 - Again, no actual return, use send to and clientaddr from main
 - DOCUMENTATION USED:
 - <https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/>
 - Start game <player> <n> <#holes>

- Initiates a game of golf with $1 \leq n \leq 3$ additional players
- <player> is dealer
- Game is parameterized by number $1 \leq \text{\#holes} \leq 9$ to player, default is 9
- Returns FAILURE if:
 - Player is not registered
 - N is not in proper range
 - Not at least n other players registered with tracker
 - Holes is not in proper range
- Query games
 - Query games of six card golf currently ongoing
 - Returns a code == number of ongoing games and a list that includes info of each game
 - Game identifier, name of dealer, names of other players
 - If no games ongoing, return 0 and an empty list
- De-register <player>
 - Removes state info about player at tracker and exit the application
 - Command returns success if and only if the given player is not involved in ongoing games
 - In the case they are, the tuple associated with the peer is deleted from database and the peer can safely exit golf game
 - If peer is involved in a game, command returns FAILURE