

Forma Bash : enoncé partie 2

On va s'intéresser ici à la communication entre un client et un serveur. Lorsque le client va lancer un jeu, il va envoyer les informations au serveur, puis la partie va se jouer sur le serveur et le résultat de la partie sera affiché pour le client. On va commencer par la mise en place du système de jeu sans communication avec le serveur.

Créez un fichier `c_l i e n t . s h`. Ce fichier prend en argument le nom du joueur et va :

- Vérifier que le joueur a déjà un compte (renvoyant un message d'erreur dans le cas contraire).
- À l'aide d'une boucle `while read variable`, demander à l'utilisateur à quel jeu il veut jouer, et si le jeu existe, il faut jouer le jeu en affichant chaque seconde la durée qu'il reste avant la fin du jeu.
- À la fin, afficher si le joueur a gagné ou non (tout en incrémentant son compteur de victoires situé dans `compte/'nom_du_joueur'/Nbr_victoire`).

On s'intéresse maintenant à la communication entre le client et le serveur, qui se fera à l'aide de tubes. Créez un fichier `s e r v e u r . s h`. Ce fichier ne prendra aucun argument et devra :

- Créer un tube nommé `s e r v e u r . t u b e` et l'ouvrir en lecture et en écriture. C'est par ce tube que le client enverra ses informations au serveur. Pour répondre au client, le serveur enverra les informations par un tube nommé `'nom_du_joueur' . s h`.
- Dans une boucle infinie, lire et placer dans les variables « `joueur` », « `donnee` » et « `jeu` » le contenu du tube `s e r v e u r . t u b e`.

Ensuite, il devra faire tout ce que `c_l i e n t . s h` faisait avant et renvoyer les résultats à `c_l i e n t . s h`.

Il faut donc modifier `c_l i e n t . s h` afin qu'il envoie au serveur (par le biais de `s e r v e u r . t u b e`) les données « joueur », « donnee » et « jeu », et qu'il attende ensuite la réponse du serveur pour l'afficher (la communication entre le serveur et un client se fera à l'aide d'un tube '`nom du client`'.`tube`).

Dans la boucle de `c_l i e n t . s h` il n'y a donc plus que :

- le test pour savoir si le jeu existe, sinon, on envoie les informations dans `s e r v e u r . t u b e` (le nom du joueur, l'action qu'il veut faire (choisie ou `En_jeu` par exemple), et le jeu auquel il veut jouer, s'il veut jouer et rien sinon)
- Et une boucle `while`, qui affiche ce que lui renvoie le serveur tant que le serveur ne renvoie pas un message pour dire qu'il a fini de traiter l'information ("Fin" par exemple).

Maintenant, complétez `s e r v e u r . s h` pour qu'il fasse ce que faisait `c_l i e n t . s h` et qu'il lui renvoie les résultats. C'est-à-dire :

- Si `$donnee == choisie`, écrire dans le shell que le joueur va jouer à un jeu, et modifier son statut (écrire "En ligne" dans `compte/$joueur/Status`).
- Si le joueur va jouer à un jeu, écrire que le joueur va jouer à `$jeu`, et calculer la durée du jeu. À l'issue du jeu, il met à jour le score du joueur s'il a gagné. Ensuite, il renvoie un message au client pour dire qu'il a fini de jouer.

Bonus :

Rajoutez la possibilité pour le joueur de voir le score des autres utilisateurs. Ainsi, à la place de choisir un jeu à jouer, il pourra écrire "score" puis le nom du joueur dont il veut savoir le score, et le serveur ira chercher le score du joueur en question pour lui renvoyer.

Quel problème cette nouvelle action peut-elle causer ?