

Jogo da Velha em Rede

Evandro Fernandes Giovanini

EP2 de MAC0448

Implementação

Cliente e servidor escritos em Python 3

Implementação

Cliente e servidor escritos em Python 3

- Servidor cria duas threads, para escutar em TCP e UDP

Implementação

Cliente e servidor escritos em Python 3

- Servidor cria duas threads, para escutar em TCP e UDP
- Cliente cria duas threads, para escutar servidor e usuário

Protocolo

Comandos simples

Protocolo

Comandos simples

- `CREATE nomeDeUsuario pass1234`

Protocolo

Comandos simples

- CREATE nomeDeUsuario pass1234
- LOGIN nomeDeUsuario pass1234

Protocolo

Comandos simples

- CREATE nomeDeUsuario pass1234
- LOGIN nomeDeUsuario pass1234
- LISTUSERS

Protocolo

Comandos simples

- CREATE nomeDeUsuario pass1234
- LOGIN nomeDeUsuario pass1234
- LISTUSERS
- HALLOFFAME

Protocolo

Comandos simples

- CREATE nomeDeUsuario pass1234
- LOGIN nomeDeUsuario pass1234
- LISTUSERS
- HALLOFFAME
- LOGOUT

Protocolo - Continuação

Mais comandos:

Protocolo - Continuação

Mais comandos:

- JOIN usuarioAconvidar

Protocolo - Continuação

Mais comandos:

- JOIN usuarioAconvidar
- JOINACCEPT usuarioQueConvidou

Protocolo - Continuação

Mais comandos:

- JOIN usuarioAconvidar
- JOINACCEPT usuarioQueConvidou
- REQPLAY 000+--+000

Protocolo - Continuação

Mais comandos:

- JOIN usuarioAconvidar
- JOINACCEPT usuarioQueConvidou
- REQPLAY 000+--+000
- PLAY 4

Protocolo - Continuação

Mais comandos:

- JOIN usuarioAconvidar
- JOINACCEPT usuarioQueConvidou
- REQPLAY 000+--+000
- PLAY 4
- FORFEIT

Protocolo - Continuação

Mais comandos:

- JOIN usuarioAconvidar
- JOINACCEPT usuarioQueConvidou
- REQPLAY 000+--+000
- PLAY 4
- FORFEIT
- REQBOARD

Manutenção do Estado

- Para cada partida existe uma thread independente

Manutenção do Estado

- Para cada partida existe uma thread independente
- Esta thread consulta o dicionário que mantém na memória a lista de usuários logados

Manutenção do Estado

- Para cada partida existe uma thread independente
- Esta thread consulta o dicionário que mantém na memória a lista de usuários logados
- Quando um usuário não responde o Heartbeat, o servidor passa a aceitar um novo login dele

Manutenção do Estado

- Para cada partida existe uma thread independente
- Esta thread consulta o dicionário que mantém na memória a lista de usuários logados
- Quando um usuário não responde o Heartbeat, o servidor passa a aceitar um novo login dele
- Este novo login substituirá o antigo no dicionário de usuários logados, e a thread da partida continua sua execução como se nada tivesse ocorrido.

Captura de tela

```
evandro@localhost:~/work/448/ep2
```

3	4	5
6	7	8

```
SERV: PING 4522
Enviado para servidor: PONG 4522
PLAY 0
Enviado para servidor: PLAY 0
> SERV: REQPLAY +00000000-
```

x	1	2
3	4	5
6	7	o

Conclusão

- EP sem nada muito complicado, porém bastante extenso
- Trabalhar com a biblioteca threading de Python foi novidade

Perguntas

Perguntas?