Démonstration PODP 2

Mathieu Campan, Besson Germain, Simon Hautesserres
Groupe L1

Département Sciences du Numérique
2022-2023

1 Premier scénario : manuel

Le but de ce scénario est de montrer le bon fonctionnement de PODP2 quand nous créons quelques clients, ici 3, à la main sur le modèle d'IRC et de vérifier la bonne propagation de l'objet, qui a été implanté dans cette partie.
Nous avons donc deux clients au commencement, de type IRC_notif, c'est à dire qu'ils sont de base abonnés à l'objet.
Lorsque le premier client décide d'écrire dans l'objet "salut1", la modification apparait dans nbNotif et l'objet est immédiatement propagé vers le deuxième client :

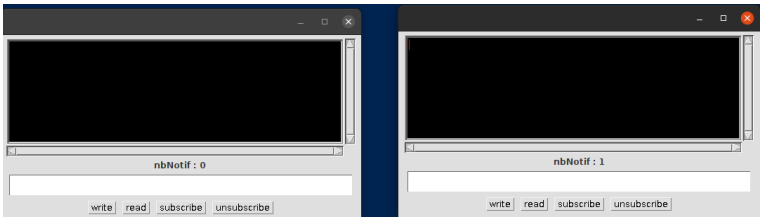

FIGURE 1 – Ecriture du premier client

Par la suite, nous décidons de faire se désabonner le deuxième client et le premier décide de réecrire. La notification n'est donc pas envoyée mais il reçoit cependant quand même la modification quand il fait une demande en lock_read comme ci-dessous :

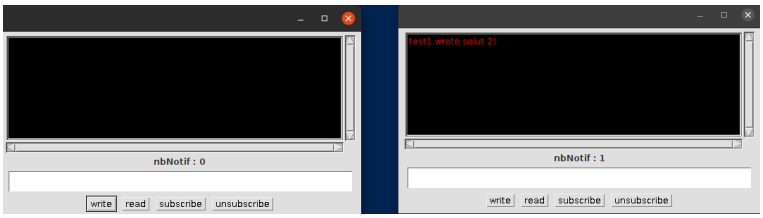

FIGURE 2 – Lecture du deuxième client

Nous remarquons d'ailleurs que nbNotif reste à 1 car il considère que le deuxième client n'est plus abonné et donc ne vérifie pas ses notifications.
Cependant, si le client se réabonne et fait de nouveau un read, la notification passe cette fois ci à zéro :

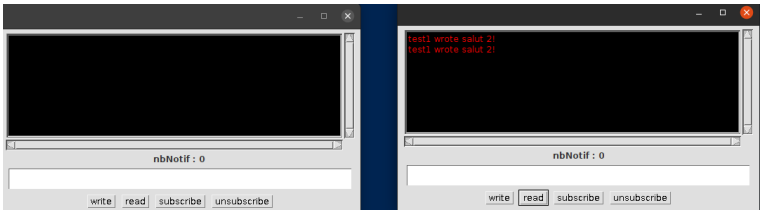

FIGURE 3 – Vérification modification notification

Rajoutons maintenant un troisième client. Celui-ci est un IRC classique, qui n'a pas les options d'abonnnements.
Il décide d'effectuer une modification et d'écrire "bonsoir". Celle ci apparait bien dans les notifications des deux autres clients :

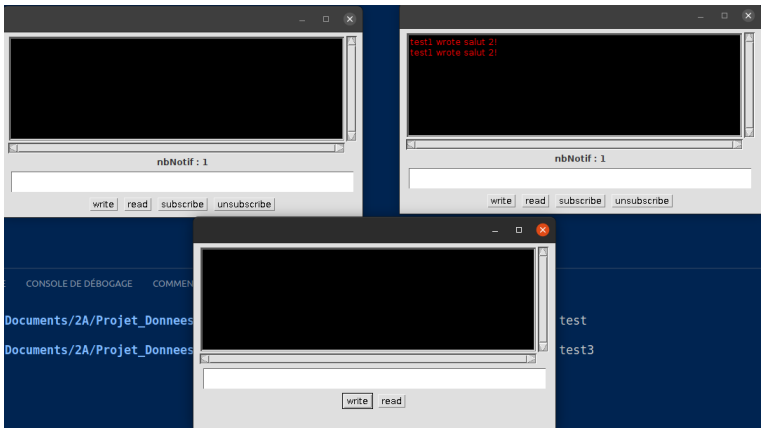

FIGURE 4 – Ajout client IRC classique

Nous décidons que les clients font de nouveau un read, la modification est toujours bien présente et la notification est réinitialisé à zéro :

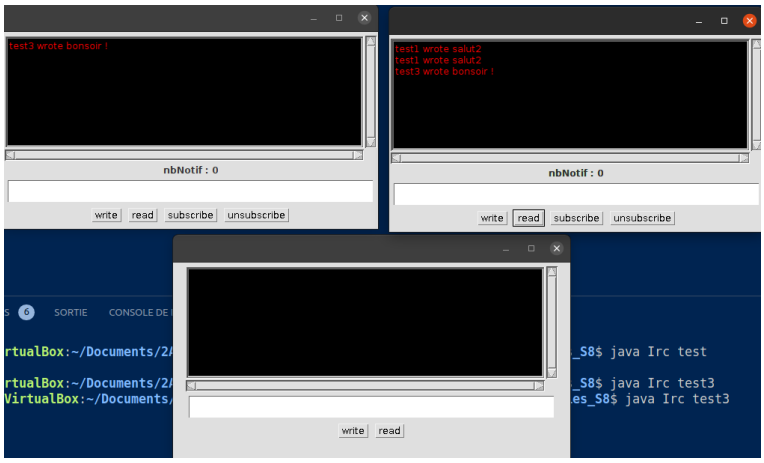

FIGURE 5 – Vérification modification IRC classique

Pour finir nous décidons de faire se désabonner les deux premiers clients et de vérifier si ils recoivent quand même la modification du troisième client, qui écrit "salut".
C'est bien le cas et nous pouvons l'observer ci-dessous :

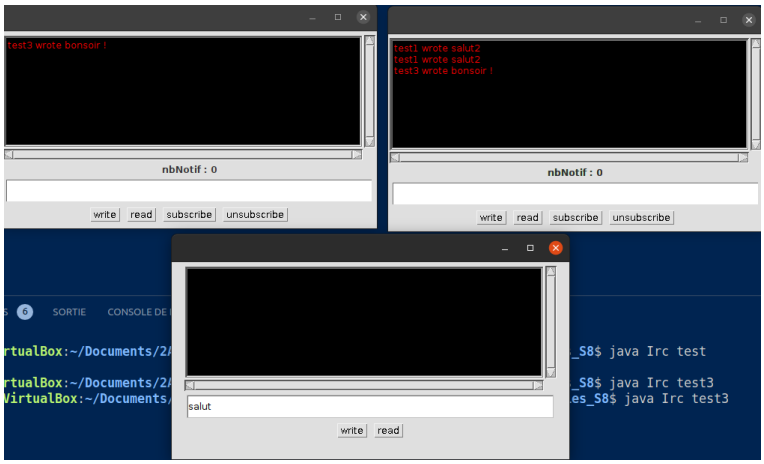

FIGURE 6 – Réabonnement des clients

2 Deuxième scénario : script sh

Afin d'automatiser les tests et vérifier les possibles blocages de notre système, nous avons adapté notre script **lancementTest.sh** qui lance un nombre défini par l'utilisateur de ClientNormal et de ClientLazy (qui garde le lock pendant un court moment) avec un abonnement automatique et un désabonnement sporadique.
Par exemple, choisissons 5 clients normaux et 2 clients lazy. Nous pouvons suivre la trace des différents locks fait pendant l'exécution dans le ficher **test.txt** et les notifications reçues dans **test_abonne.txt** :

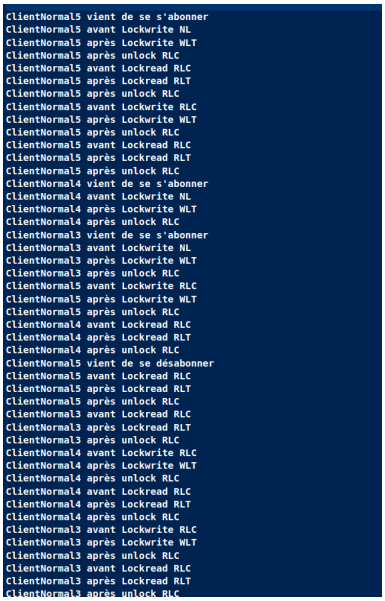

FIGURE 7 – Suivi des locks dans test.txt

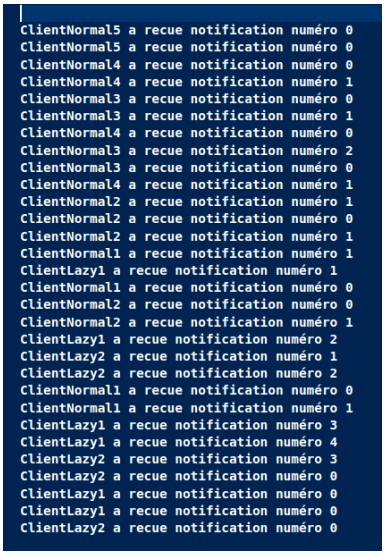

FIGURE 8 – Suivi des notifications dans test_abonne.txt

Celle ci est cohérente par rapport aux changements de lock que l'on souhaite, notamment qu'après un lock_write, les verrous passent à RLC et pas WLC comme dans la version précédente.
De plus, nous voyons que lorsque ClientNormal 5 se désabonne, il ne reçoit plus de notifications mais les verrous restent cohérents.