**Question DS écrit java**

**4x Expliquez ce qu’est un code ré entrant ou thread safe.**

Un code thread safe est un code capable de fonctionner correctement alors qu’il est exécuté simultanément au sein du même espace d’adressage par plusieurs thread.

**4x Donnez un exemple de code C ou en java qui ne soit pas ré entrant et pourquoi ?**

Une image contenant texte

Description générée automatiquementNon thread safe, car pas synchroniser

**4x Donnez une solution pour rendre votre code thread safe**

Une image contenant texte

Description générée automatiquement

Ajouter un synchronised

**3x Un client envoie un datagramme UDP à un serveur. Vous avez réalisé une capture du paquet IP correspondant à cet envoi (capture en héxadécimal).**

Une image contenant texte

Description générée automatiquement

**Sur quel port écoute le serveur (réponse en décimal) ? Justifiez votre réponse. Quel est le port du client (réponse en décimal) ? Justifiez votre réponse.**

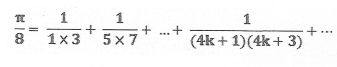
Nous savons que la partie en rouge sur le datagramme correspond à l’entête IP composé de 20 octets. 7F 00 00 01 est l’IP source et 7F 00 00 01 est l’IP destinataire. La partie en vert correspond au 8 octets de l’entête UDP. 0405 correspond au port source et 0A05 correspond au port destinataire.

A05 = 2565 et 405 = 1029.

**3x Quel est le numéro de port pour un serveur http ? Qui est 127.0.0.1 ?**

Le numéro de port pour un serveur http est le port 80. 127.0.0.1 est l’adresse localhost ce qui signifie qu’il s’agit de l’interface de la machine local.

**2x D’après la formule de Leibniz, on sait que :**



**Sur une machine 4 processeur écrivez un programme en Java calculant cette suite le plus rapidement possible. Avec k = 100 000 000. Votre programme devra afficher la valeur de PI calculée, et le temps mis pour faire ce calcul.**

Une image contenant texte

Description générée automatiquement

**2x Expliquez dans quel cas un programme se termine avec une erreur « segmentation fault ».**

Une erreur segmentation fault signifie qu’un processus essaie de lire ou d’écrire la mémoire d’un autre processus.

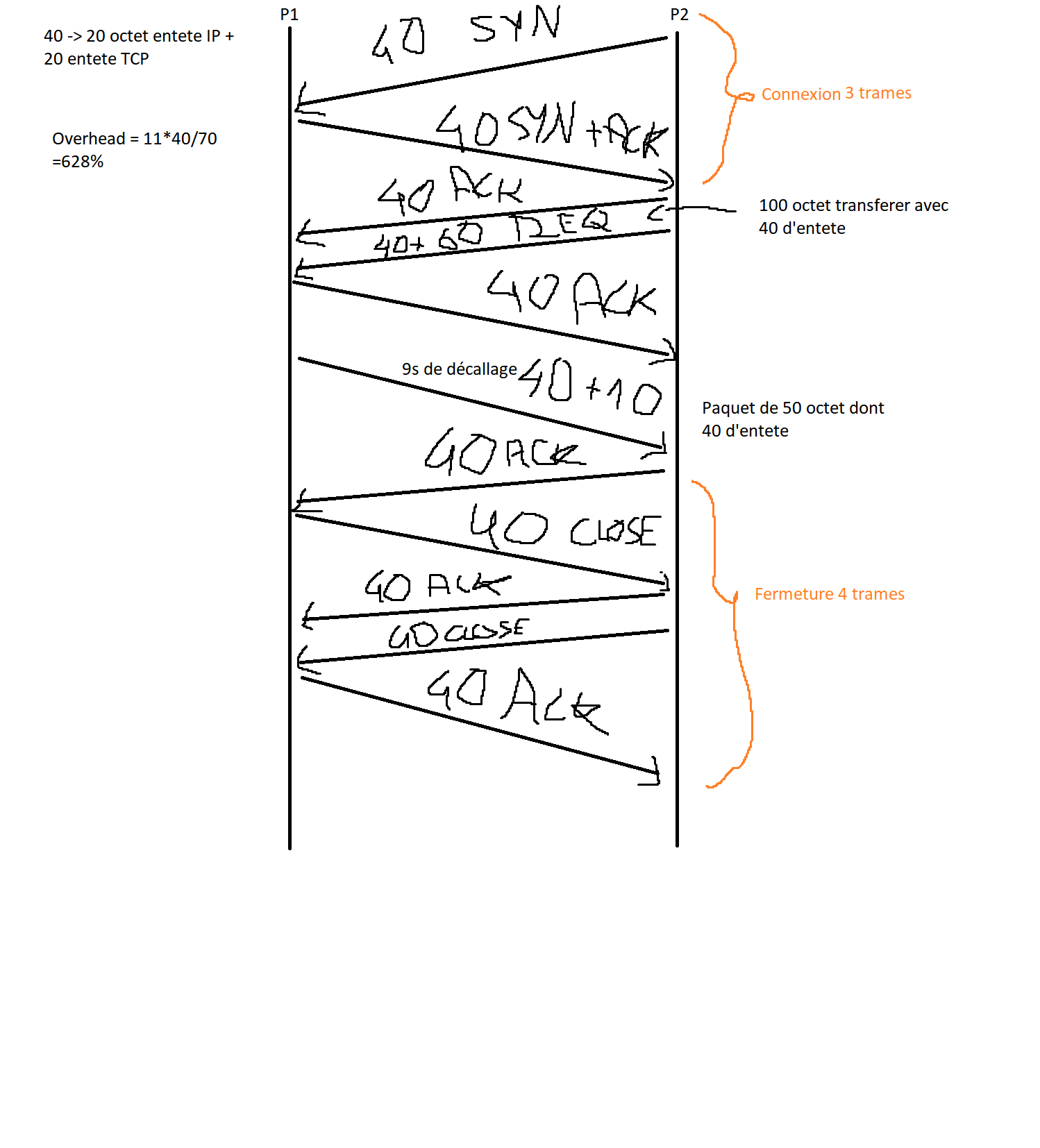
**2x Nous considérons deux processus P1 et P2 communiquant entre eux par TCP. P1 est exécuté sur la machine A, P2 sur la machine B. P1 joue le rôle du serveur, P2 joue le rôle du client.**

Une image contenant texte

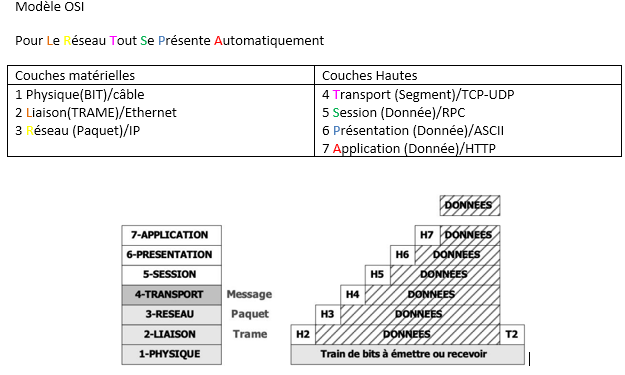
Description générée automatiquement

**Décrivez tous les paquets IP échangés entre P1 et P2 (taille, contenu). Faites un calcul de l’overhead apporté par les couches 3 et 4 du modèle OSI.**

Entête TCP 20 octets et IP 20 octet et UDP 8 octets



**2x Indiquez par un dessin les 7 couches du modèle OSI. Pour les couches 1 à 4 donnez un exemple d’implémentation**



1. Trains de bits (BITS) wifi
2. Adresse physique (TRAME) can
3. Détermine le parcours et l’adressage logique IP (Paquet) IP
4. Connexion de bout en bout et contrôle de flux TCP (Segment) TCP UDP

**1x Expliquez le fonctionnement de l’API des sockets pour TCP**

1. Le serveur se déclare auprès de la couche transport
2. Phase de connexion entre le premier client et le serveur
3. Acceptation du premier client par le serveur
4. Echange de données bidirectionnelle entre le 1er client et le serveur
5. Terminaison de la connexion entre le 1er client et le serveur

**1x Donnez 5 techniques permettant d’implémenter un serveur parallèle. Donnez des exemples de logiciels utilisant ces techniques pour au moins 3 techniques. Détaillez le fonctionnement de chaque technique.**

1. **Serveur multi processus** -> Le travail est délégué à un processus fils crée via fork().
2. **Serveur multi processus** **avec réservoir de processus** -> processus père gère un réservoir de processus fils avec une stratégie de nombre moyen de fils. Le travail est délégué au fils inoccupé.
3. **Serveur multi threadé standard** -> Utilise des threads à la place des processus. Le travail est délégué à un thread fils crée via new Thread().
4. **Serveur multi threadé avec réservoir de processus** -> le thread père gère un réservoir de thread fils avec une stratégie de gestion de fils moyen. Le travail est délégué au fils inoccupé.

**Exemple : TOMCAT**

1. **Serveur parallèle classique** -> Une tâche attend les demandes de connexion qui les vérifie valide la demande puis la transfère à une tâche fille.

**1x Quelle est l’année d’invention du WEB ?**

Le WEB a été inventé en 1989.

**1x Expliquez les termes little endian et big endian**

Little endian correspond à l’écriture de l’octet de poids faible à l’adresse la plus petite (envoie de l’octet de poids faible en premier).

Big endian correspond à l’écriture de l’octet de poids fort à l’adresse la plus petite (envoie de l’octet de poids fort en premier).

**1x Expliquez comment est codé le nombre 770 dans un entier 32bits dans chaque cas (little et big endian) Pour chaque octet vous donnerez sa valeur en décimal.**

Décomposition de 770 en puissance de 2 pour le big endian. On décompose donc 1\*2^1+1\*2^8+1\*2^9 ce qui nous donne en HEXA 0x302. Pour passer de big endian à little endian il suffit de permuter le premier et le dernier octet. Nous obtenons donc 0x203 ce qui correspond à 770 en décimale.

**1x Que signifie « network byte ordre » ? Pourquoi a-t-on du inventer cette notion ?**

Network bit order correspond au format big endian du TCP IP. Nous avons dû inventer cette notion, car les deux standards existent en informatique (big et little endian) et pour éviter des problèmes de communications il est important que tout le monde parle le même langage.

**1x Si un paquet IP est perdu lors d’un échange par UDP, quel est l’impact sur votre programme ? En tant que programmeur, qu’avez-vous à faire par rapport à la perte d’un paquet IP si vous utilisez UDP ?**

Le paquet est perdu et ne sera pas envoyé renvoyer. Il faut donc implémenter une gestion d’exception qui demande le renvoi de la trame.

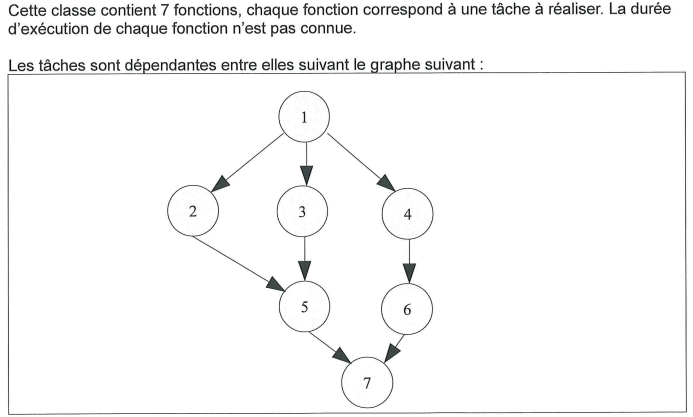
**1x Si un paquet IP est perdu lors d’un échange par TCP, quel est l’impact sur votre programme ? En tant que programmeur, qu’avez-vous à faire par rapport à la perte d’un paquet IP si vous utilisez TCP ?**

Le paquet sera envoyé dans une trame ultérieure. Il faut donc vérifier que la trame soit complète, si ce n’est pas le cas il faut attendre la réception de la fin de la trame avant de traiter les données reçues.

**1x On vous fournit le code suivant :**

Une image contenant table

Description générée automatiquement

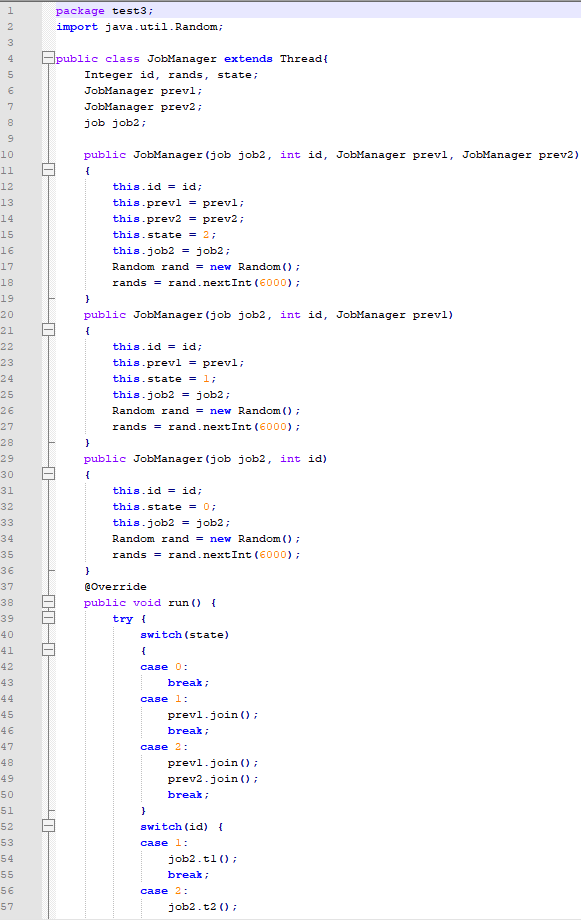


Une image contenant texte, intérieur, capture d’écran

Description générée automatiquement

Une image contenant texte

Description générée automatiquement



Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement