

PROJET
STRUCTURE DE DONNEES ET ALGORITHMES
Gestion d'une messagerie

Le projet consiste à développer une application de gestion de messagerie.

Un message, adressé par un expéditeur à un destinataire, est transmis via le réseau par un ensemble de paquets-réseau, chacun contenant un paquet de données de taille fixe du message transmis. Le nombre de paquets est lié à la fragmentation du message en blocs de données de taille t caractéristique du réseau. Un paquet-réseau est constitué d'un entête regroupant ses informations propres (-expéditeur, -destinataire, -temps d'émission, -indicateur de fin de message, -numéro du bloc de données) et de son bloc de données (constitué de t caractères) propre au message transmis. Chaque message est d'une longueur quelconque (i.e., constitué d'un nombre quelconque de blocs de données).

La communication réseau fait qu'à la réception : les paquets d'un même message vont être entrelacés avec des paquets-réseau d'autres messages (autres expéditeurs, autres destinataires). De plus, l'ordonnancement séquentiel des paquets d'un même message n'est pas garanti à la réception : le paquet numéroté n peut être reçu avant le paquet numéroté $n-1$. Des pertes de paquets peuvent se produire dans la communication réseau, la réception de tous les paquets constituant un message n'est par conséquent pas garantie.

Les messages arrivent tous sur une même ligne de transmission. Le rôle de la messagerie est donc d'organiser en mémoire les paquets reçus de telle sorte que dès la réception de l'ensemble des paquets d'un message, ce message soit reconstitué (suivant l'ordre séquentiel de ses blocs de données), affiché à l'écran et archivé dans la boîte des messages (*mailbox*) du destinataire. On considérera une perte de paquet si un message en cours de réception ne reçoit aucun paquet dans un délai de temps τ . Ce temps est un paramètre de l'application, il est estimé par le nombre de paquets traités par la messagerie depuis le dernier paquet traité du message. Dans le cas d'une perte de paquet détectée, le message sera abandonné et supprimé en mémoire. Les principales opérations de gestion de la messagerie liées aux numéros des paquets-réseau analysés de la ligne de transmission (-réception d'un nouveau message, -archivage de message dans une *mailbox*, -abandon de message dû à une perte de paquet, -abandon des messages en cours de réception lors de la fin de transmission) seront tracées dans un journal (*log file*).

La ligne de transmission est simulée par un fichier texte. Un paquet-réseau transmis correspond à deux lignes du fichier :

(1) la première ligne correspond à l'ordre de réception des paquets-réseaux depuis le début de la transmission et à l'entête du paquet-réseau dont les champs d'information sont les suivants : -expéditeur, -destinataire, -temps d'émission (date et heure), -indicateur de fin de message, -numéro du bloc de données,

(2) la deuxième ligne correspond au bloc de données (t caractères).

Format : les différents champs sont séparés par un espace, l'expéditeur et le destinataire sont des identificateurs (chaîne de caractères sans délimiteur de chaîne), le temps est donné au format *jj/mm/aa hh:mm:ss*, l'indicateur de fin de message est à sémantique booléenne (0 pour faux et 1 pour vrai), le numéro du bloc de données est un entier naturel >0 .

Ci-joint, un exemple de simulation de la ligne de transmission avec $t=20$.

En fonction du fichier de simulation de la ligne de transmission, les résultats attendus de l'application de messagerie demandée sont constitués des traces suivantes : -l'affichage à l'écran des messages dès leur réception dans leur intégralité, -la trace des *mailbox* et -la trace du fichier *.log*.

```
1 caraty gr9a12 26/11/17 14:16:37 0 1
Commentez vos source
2 caraty gr1a4 26/11/17 14:20:02 0 2
de commenter les fi
3 caraty gr5a8 26/11/17 14:35:23 0 1
Programmez sans nomb
4 caraty gr1a4 26/11/17 14:20:02 0 1
Il est indispensable
5 caraty gr5a8 26/11/17 14:35:23 1 4
ex. 255, ".txt").
6 caraty gr1a4 26/11/17 14:20:02 0 4
uls fichiers à consu
7 caraty gr1a4 26/11/17 14:20:02 0 3
chiers d'entête : se
8 caraty gr5a8 26/11/17 14:35:23 0 3
ntes littérales, par
9 caraty gr1a4 26/11/17 14:20:02 0 5
lter pour la réutili
10 caraty all 26/11/17 15:27:57 0 1
Bon courage à tous..
11 caraty gr1a4 26/11/17 14:20:02 1 6
sation.
12 caraty all 26/11/17 15:25:41 1 1
Bon travail...
13 caraty gr5a8 26/11/17 14:35:23 0 2
res magiques (consta
14 caraty gr9a12 26/11/17 14:16:37 1 2
s.
15 caraty all 26/11/17 15:27:57 1 2
. MJC
```

```
caraty gr1a4 26/11/17 14:20:02
Il est indispensable de commenter les
fichiers d'entête : seuls fichiers à
consulter pour la réutilisation.
```

```
caraty all 26/11/17 15:25:41
Bon travail...
```

```
caraty gr5a8 26/11/17 14:35:23
Programmez sans nombres magiques
(constantes littérales, par ex. 255,
".txt").
```

```
caraty all 26/11/17 15:27:57
Bon courage à tous... MJC
```

Trace d'exécution attendue à l'écran

```
Mailbox all
caraty all 26/11/17 15:25:41
Bon travail...
caraty all 26/11/17 15:27:57
Bon courage à tous... MJC
```

```
Mailbox gr1a4
caraty gr1a4 26/11/17 14:20:02
Il est indispensable de commenter les
fichiers d'entête : seuls fichiers à
consulter pour la réutilisation.
```

```
Mailbox gr5a8
caraty gr5a8 26/11/17 14:35:23
Programmez sans nombres magiques
(constantes littérales, par ex. 255,
".txt").
```

Les *mailbox* devront suivre le format donné en exemple. Leur nom est construit à partir de l'identificateur du destinataire et de l'extension ".mb" Dans notre exemple : all.mb, gr1a4.mb, gr5a8.mb

Pour l'exemple de ligne de transmission donné avec un délai $\tau=10$, le journal (messagerie.log) attendu est donné ci-dessous. Dans ce fichier, toutes les opérations de gestion dont on conserve l'historique (-détection de nouveau message, -détection de fin de message, -archivage mailbox, -perte de paquet, -fin de transmission) sont repérées par l'ordre de réception dans la transmission du paquet-réseau analysé et respectent le format suivant :

```
1 Détection de nouveau message caraty gr9a12 26/11/17 14:16:37 0 1
2 Détection de nouveau message caraty gr1a4 26/11/17 14:20:02 0 2
3 Détection de nouveau message caraty gr5a8 26/11/17 14:35:23 0 1
5 Détection de fin de message caraty gr5a8 26/11/17 14:35:23 1 4
10 Détection de nouveau message caraty all 26/11/17 15:27:57 0 1
11 Détection de fin de message caraty gr1a4 26/11/17 14:20:02 1 6
11 Archivage mailbox gr1a4 26/11/17 14:20:02
11 Perte de paquet, suppression de message caraty gr9a12 26/11/17 14:16:37
12 Détection de nouveau message caraty all 26/11/17 15:25:41 1 1
12 Détection de fin de message caraty all 26/11/17 15:25:41 1 1
12 Archivage mailbox all 26/11/17 15:25:41
13 Archivage mailbox gr5a8 26/11/17 14:35:23
14 Détection de nouveau message gr9a12 26/11/17 14:16:37 1 2
14 Détection de fin de message gr9a12 26/11/17 14:16:37 1 2
15 Détection de fin de message caraty all 26/11/17 15:27:57 1 2
15 Archivage mailbox all 26/11/17 15:27:57
Fin de transmission, suppression de message caraty gr9a12 26/11/17 14:16:37
```

Cadre du développement logiciel

Le projet est à réaliser en binôme. Les membres d'un binôme seront de préférence du même groupe.

Vous devez programmer et tester l'application de gestion d'une messagerie en respectant toutes les spécifications fonctionnelles données.

L'un des objectifs de ce projet est la réutilisation. Vous devez choisir et réutiliser des composants techniques (tableau dynamique, pile, file, liste, ...), vus en cours ou en travaux pratiques. Vous choisirez les composants les plus adaptés à la résolution des problèmes posés par l'application. De plus, vous veillerez à la bonne gestion de la mémoire dynamique.

Votre travail sera évalué à partir du dossier de développement logiciel de l'application demandée et dont les spécifications sont données ci-dessous.

Dossier du développement logiciel

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est déterminante pour l'évaluation de votre travail. La composition de votre dossier doit être la suivante :

- Une page de garde indiquant le nom et le groupe des membres du binôme, l'objet du dossier. Une table des matières de l'ensemble du dossier (incluant les annexes) avec la pagination de toutes les rubriques. L'ensemble du dossier doit être paginé (dont les sources et les annexes).

- Une brève présentation du projet (fonctionnalités et résultats de l'application).
- Le graphe de dépendance des fichiers sources de votre application. Tous les composants (réutilisés ou développés) de l'application devront figurer sur le graphe (cf. Cours 4).
- Le rapport de développement par sprints (cf. Cours Projet) avec le bilan de validation des tests pour chacun des sprints. Pour chaque sprint, donnez : (1) le(s) jeu(x) de données de test (JDT) de votre application (fichier(s) texte(s) de la ligne de transmission) et le(s) trace(s) d'exécution de votre programme, (3) le bilan de validation (comparaison des résultats observé(s) et attendu(s) concluant au 0-défaut ou non). Remarque : un JDT personnel doit être commenté de manière à illustrer le test effectué.
- Un bilan du projet : votre retour d'expérience regroupant les difficultés rencontrées, ce que vous avez appris à maîtriser, ce qui est réussi et/ou peut être amélioré.
- En annexe au dossier donnez le listing complet de vos sources. La présentation doit suivre l'ordre suivant : (1) le sommaire paginé récapitulatif vos sources, suivi des sources dans l'ordre suivant (2) le point d'entrée de l'application (programme principal), (3) les fichiers de spécification (.h) des composants, (4) les fichiers corps (.cpp) des composants. Vous listerez en fin les composants techniques.

Remarque : Vous trouverez sous COMMUN dans le répertoire projet l'exemple donné de fichier de ligne de transmission et les résultats attendus. Ce JDT fait partie de vos tests, vous vérifierez que les résultats de votre application sont bien les résultats attendus. Mais ces tests sont insuffisants pour bien tester votre application, à vous de les compléter.

Spécifications

Le **code source** doit être **structuré en composants** et **documenté**. En particulier, les fichiers, les fonctions et toute structure de données doivent être documentés suivant le formalisme Javadoc. Les **préconditions** des fonctions doivent obligatoirement être **documentées** et **vérifiées** par assertion. Vous veillerez à la **bonne gestion** de la **mémoire dynamique** dans votre programme.

Suivez toutes les **spécifications données** sous peine de pénalisation à l'évaluation. Entre autres, suivez toutes les spécifications données pour la rédaction du dossier de développement logiciel.

Recette et date limite de remise de projet

Vous passerez la **recette** de votre projet la semaine du **8 janvier 2018 (en deuxième séance de TP)**. La date limite de remise de votre dossier de développement (version papier et version électronique) est fixée le **lundi 15 janvier 2018** au secrétariat. Ce même jour, vous déposerez la version électronique de votre projet dans le **puits** : l'archive (.zip) de nom **Nom1Gr?Nom2Gr?.zip** constituée de **tous les sources (sans l'exécutable .exe)** de votre projet et du dossier de développement logiciel (.pdf).

ATTENTION : TOUT PROJET PLAGIE SERA SANCTIONNÉ PAR LA NOTE DE 0/20.