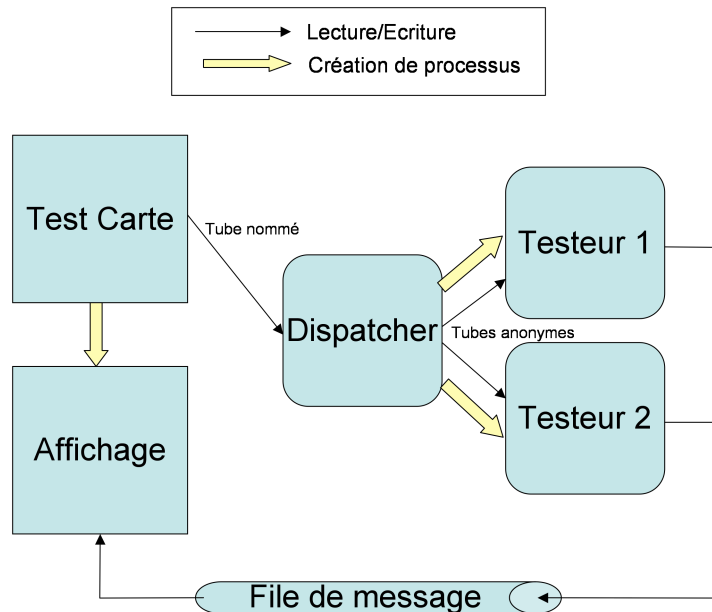


OS Réseaux et Programmation Système

TD3

On veut créer une application qui permet de vérifier la validité de couples $\langle \text{numéro_de_carte_de_credit}, \text{date_expiration} \rangle$. L'architecture de l'application est la suivante :



Rôle des processus :

- Test Carte : récupère les informations sur les cartes de crédit à tester et les envoie au Dispatcher
- Dispatcher : lit les informations envoyées par Test Carte et les répartit entre les différents testeurs disponibles
- Testeur : vérifie les informations de cartes envoyées par le Dispatcher et poste le résultat (numéro de carte et validité) dans une file de message
- Affichage : lit les messages envoyés sur la file et affiche les résultats.

On propose le prototype de programme suivant pour réaliser l'application (voir page 2 et 3). Complétez le afin de réaliser l'application demandée.

test_carte.c

```
#define CLE 312
```

```
struct message{  
    // à compléter 1  
}
```

```
main() {
```

```
    struct message mess;
```

```
    int pid;           // identifiant processus
```

```
    int fileid;       // identifiant de la file de message
```

```
    int tubenommeid;  // identifiant du tube nommé
```

```
    char numcarte[16]; // numéro de carte bancaire
```

```
    char date[4];     // date d'expiration
```

```
    char pidaff[10];  // identifiant du processus affichage
```

```
    mode_t mode;      // paramètres d'ouverture du tube nommé
```

```
    mode = S_IRUSR | S_IWUSR;
```

```
    pid = fork();
```

```
    if (// à compléter 2){
```

```
        // à compléter 3, initialisation communication
```

```
        while(true){
```

```
            // à compléter 4, communication
```

```
        }
```

```
    }else {
```

```
        // à compléter 5, initialisation communication
```

```
        while(true){
```

```
            getInformationCarte(&carte, &expire);
```

```
            // à compléter 6, communication        }
```

```
        }
```

```
    }
```

dispatcher.c

```
#define CLE 312
```

```
struct message{  
    // à compléter 7  
}
```

```
main() {
```

```
    struct message mess;
```

```
    int pid1, pid2;           // identifiants processus  
    int pip1[2], pip2[2];    // identifiants tubes anonymes
```

```
    int fileid;              // identifiant de la file de message  
    int tubenommeid;        // identifiant du tube nommé
```

```
    char numcarte[16];       // numéro de carte bancaire  
    char date[4];           // date d'expiration  
    char pidaff[10];        // identifiant du processus affichage
```

```
    bool fils1;              // booléen pour test dispatcher  
    fils1 = 1;
```

```
    // à compléter 8, initialisation communication
```

```
    pid1 = fork();  
    if (pid1 == 0){
```

```
        // à compléter 9, initialisation communication
```

```
        while(true) {
```

```
            // à compléter 10, communication
```

```
            // testCarte(numcarte, date); renvoie 1 si la carte est valide, 0 sinon
```

```
        }
```

```
    }else {
```

```
        pid2 = fork();
```

```
        if (pid2 == 0){
```

```
            // à compléter 11, initialisation communication
```

```
            while(true) {
```

```
                // à compléter 12, communication
```

```
                // testCarte(numcarte, date); renvoie 1 si la carte est valide, 0 sinon
```

```
            }
```

```
        }else {
```

```
            // à compléter 13, initialisation communication
```

```
            while (true) {
```

```
                // à compléter 14, communication
```

```
            }
```

```
        }
```

```
    }
```

```
}
```