Cahier des charges : projet stage

Création d'une application d'exercices de systèmes et codage de l'information

Client : GENIET Annie

Développeurs : Kone Yacouba, Akoury Joseph, DIOP Amadou Tidiane, DISSOU Kefa, IRANNEZAHAD Avin, KAFABA Alimou, TENDOMENE Ora, VERDIER Liam.

Année : 2022 / 2023

Contexte

La troisième année de licence Génie Bio-informatique à l'université de Poitiers comporte plusieurs aspects d'apprentissage de l'informatique :

- Une initiation à l'informatique où sont vus les principes du codage de l'information.
- Un enseignement comportant les principes généraux des systèmes d'exploitation.

Tous ces enseignements sont vus durant les cours magistraux et mis en pratique à travers des exercices proposés lors des séances de Travaux Dirigés ou dans les Devoir maison. Ils sont nécessairement en nombre limité. Pour certains, ils sont également mis en pratique durant des Travaux Pratique.
La cliente, Mme Annie Geniet, en charge de ces enseignements, souhaite proposer aux étudiants une application leur donnant accès à un plus large panel d'exercices, afin de leur permettre de s'entraîner autant que nécessaire, tout cela ayant pour but d'acquérir les notions vues lors des cours.

Objectifs généraux :

L'application à développer comportera 4 grandes parties, chacune associée à l'un des 4 thèmes suivants :

1. Codage de l'information
2. Ordonnancement
3. Gestion de la mémoire
4. Gestion de fichiers

Chaque thème est ensuite divisé en plusieurs exercices, qui seront décrits dans les parties dédiées dans ce document. Tous les exercices suivront la même trame :

a. Les données pourront être, au choix de l'utilisateur, soit saisies soit générées aléatoirement par l'application. Dans la suite du document, le terme "entrée" représente l'une ou l'autre des solutions.

b. L'utilisateur devra saisir ses réponses dans les espaces dédiés à cet effet.

c. Si la/les réponse(s) fournie(s) est/sont correcte(s), un message apparaîtra. Dans le cas contraire, l'utilisateur pourra proposer une/des nouvelle(s) réponse(s). Il aura le droit à 3 essais au bout desquels la bonne réponse lui sera donnée.

d. Un contrôle syntaxique est mis en place, qui vérifie si les données saisies sont correctes. Un message sera affiché si la saisie n'est pas du bon type ou bien si les données ne respectent pas les contraintes. Dans le cas où la saisie est syntaxiquement correcte alors elle pourra (et seulement dans ce cas), être validée.

e. Les valeurs pourront être fournies entrecoupées d'autant d'espace que voulu.

Elles doivent donc être composées uniquement des symboles autorisés pour la base choisie et d'espaces. Sinon, elles sont considérées comme syntaxiquement incorrectes.

f. Lorsque cela est pertinent, des rappels méthodologiques seront affichés au début de l'exercice.

A la fin de l'exercice, la liste des exercices du thème est reproposée jusqu'à ce que l'utilisateur décide de revenir à l'écran où il peut choisir les thèmes.

Les parties suivantes détaillent le contenu de chacune des parties.

I. La partie Codage de l'information :

Dans l'interface principale de l'application, Si on choisit le thème Codage de l'information un nouvel écran s'affiche proposant une liste d'exercices disponibles. L'utilisateur choisit alors l'exercice qu'il veut faire.

Huit exercices sont proposés :

1. Les entiers non signés
2. Les opérations en binaire
3. Les multiplicités en binaire
4. Opérations sans calcul
5. Les entiers signés
6. Les décimaux
7. Les réels
8. Les tableaux

1) Entiers non signés

Objectif : Faire des conversions entre différentes bases Les

bases considérées sont les bases : 10-2-8-16

Données entrées : base de départ - base d'arrivée
Si les bases sont saisies, l'entier à convertir est entré
Si les bases sont générées aléatoirement, la valeur de l'entier à convertir est également générée.

Réponse attendue : le nombre codé dans la base d'arrivée.

Contraintes :

- Les entiers en base 10 sont compris entre 0 et 10 000
- Si on travaille avec les bases 2, 8 ou 16 (sans aller de ou vers la base 10), les contraintes de taille lors des entrées sont :
  - Base 2 : maximum 32 bits
  - Base 8 : maximum 10 symboles
  - Base 16 : maximum 8 symboles

Rappels méthodologiques :

- pour aller de base 10 à la base 2 : méthode de la division ou méthode de la soustraction
- pour aller de base 2 à base 8 ou 16 : méthode de la compression - pour aller de base 8 ou 16 à base 2 : méthode de l'expansion
- pour aller de base 8 à base 16 ou réciproquement : passer par la base 2 en intermédiaire.

2) Opérations en binaire

Objectif : Effectuer des opérations en base 2.

Les opérations considérées sont les multiplications, les additions et les soustractions.

Données entrées : choix de l'opération - valeur des opérandes.

Si le choix de l'opération est saisi, la valeur des opérandes est entrée. Si le choix de l'opération est aléatoire, la valeur des opérandes est également générée.

Réponse attendue : le nombre en base de 2 résultant de l'opération.

Contraintes :

- Les entrées ne pourront pas excéder 16 bits.

Rappel méthodologique :

- N'oubliez pas que : 1 + 1 = 10.
- N'oubliez pas les retenues.

3) Multiplicité

Objectif : savoir si un nombre binaire est multiple d'une puissance de 2.

Données générées : Entiers codés en binaire - un entier qui est un puissance de 2.

10 valeurs sont générées aléatoirement, ainsi que la puissance de 2.

Réponse attendue : Les valeurs qui sont des multiples de l'entier doivent être cochées.

Contraintes :

- Les entrées ne pourront pas excéder 32 bits.

- Au moins 3 multiples de la puissance de 2 donnée se trouvent parmis les 10 valeurs générées.

Rappel méthodologique :

- Si un entier est multiple d'une puissance (n) de 2, il doit se terminer par (n) 0.

4) Opérations sans calcul

Objectif : Multiplier ou diviser un nombre en base 2 par une puissance de 2.

Données entrées : Type d'opération - Nombre en base 2 - Puissance de 2. Lorsque le type d'opération est généré, les valeurs du nombre en base 2 et de l'entier en base 10 sont générées.

Réponse attendue : le nombre en base 2 résultant de l'opération.

Contraintes :

- Les base 2 ne peuvent pas excéder 16 bits en entrée
- Les entiers en base 10 sont compris entre 2 et 256 uniquement en puissance de 2
- Le nombre en base 2 doit se terminer par au moins autant de 0 que l'exposant de la puissance de 2 dans le cas de la division

Rappel méthodologique :

- Pour multiplier il faut ajouter des 0 à la fin du nombre en base 2 donné • Pour diviser il faut supprimer des 0 à la fin du nombre en base 2 donné.

5) Les entiers signés

a) Les signes

Objectif : déterminer le signe d'un entier codé en SVA/C2.

Données générées : un entier codé soit en SVA, soit en C2.
La valeur est générée aléatoirement ainsi que le format (SVA ou C2).

AG

Réponse attendue : Le signe de l'entier donné. Il doit être coché (positif/négatif).

Contraintes :

- L'entier ne peut pas excéder 16 bits.
- Dans cet exercice, une seule tentative sera accordée à l'utilisateur.

b) Conversion :

Objectif : Faire des conversions entre différents formats. Les formats considérés sont la base 10, SVA et C2.

Données entrées : format de départ - format d'arrivée

Si le format de départ est saisi, le format d'arrivée est saisi.

Si le format de départ est généré aléatoirement, le format d'arrivée est aussi généré.

Si on va vers SVA ou C2, on précise le nombre de bits sur lequel on veut faire le codage.

Réponse attendue : le nombre codé dans le format d'arrivée.

Contraintes :

- L'entier ne peut pas excéder 16 bits.
- Si on donne un entier en base 10, et le nombre de bits, alors la valeur absolue doit être comprise entre 1-2^(n-1) et 2^(n-1)-1.

Rappels Méthodologiques :

- Si l'entier codé en SVA/C2 commence par 1 : son signe est négatif.
- Si l'entier codé en SVA/C2 commence par 0 : son signe est positif.
- Pour convertir un nombre négatif de C2 vers SVA : on soustrait 1, puis on inverse les bits en faisant attention au signe.
- Pour coder un entier négatif en C2 : on part de la valeur absolue, on inverse les bits, puis on ajoute 1.

6) Décimaux

Objectif : Convertir un nombre décimal de la base 10 vers la base 2 ou inversement.

Données entrées : base de départ - base d'arrivée - nombre de bit après la virgule pour la base 2.

Si l'ordre de conversion est choisi, la valeur dans la base de départ est entrée.

Si l'ordre est choisi aléatoirement, la valeur est également générée aléatoirement.

Réponse attendue : Le nombre codé dans le format d'arrivée.

Contraintes :

- Si la valeur de départ est en base 2 elle ne peut pas dépasser 32 bits.
- Si la valeur est en base 10 elle est comprise entre 0 et 10 000.
- Si la valeur est en base 2, il y a au plus 5 bits après la virgule.
- Si on convertit en base 2, le nombre de bits après la virgule est compris entre 1 et 6.

Rappel méthodologique :

- Pour obtenir la partie décimale, on utilise la méthode de la multiplication.

7) Les réels

Objectif : Convertir un réel au format IEEE 32 bits donné en hexadécimal(ou l'inverse)
Bases considérées : 10 - 16

Données entrées : le sens de la conversion - le réel à convertir.
Le réel à convertir est, soit en format décimal (si c'est la base 10), soit en hexadécimal (si c'est en IEEE).

Réponse attendue : Le nombre codé dans le format d'arrivée.

Contraintes :

- Les valeurs en base 10 sont compris entre -10 000 et 10 000

Rappels méthodologiques :

- 1 bit de signe
- 8 bits d'exposant biaisé (biaisé de 127)
- 23 bits de mantisse.
- Ne pas oublier le bit implicite.

8) Tableau

Objectif : Calculer le nombre de mots d'un tableau et savoir les placer.

Données entrées : taille du tableau (en nombre de case) - taille d'une case (en nombre de mot) - taille d'un mot (en bits ou en octets) - adresse du premier mot du tableau.
Les valeurs sont entrés (saisies ou générées, les deux sont envisageables). Dans un second temps, le numéro de case est généré pour la deuxième question.

Réponse attendue :

- le nombre de mots contenus dans le tableau et les numéros du premier et du dernier mot du tableau.
- les mots contenus dans la case dont le numéro est généré.

Contraintes :

- La taille du tableau est comprise entre 30 et 300 cases.
- Taille mot * taille case ne doit pas dépasser 950 mots

Validé jusqu'ici
par le client Annie Geni*Rappel méthodologique :*

- Si a est le premier mot du tableau, et si une case contient n mots, le

premier mot de la case n°k est le mot a+(k-1)*n.

II. La partie ordonnancement

Objectif: Simuler l'exécution d'une famille de processus selon différents

algorithmes d'ordonnancement.

Interface :

Si on choisit le bouton\thème "Ordonnancement" un nouvel écran s'affiche, un exercice s'ouvre avec un menu déroulant permettant de choisir l'algorithme d'ordonnancement souhaité.

Huit algorithmes sont proposés :

1. Tourniquet

.2. FIFO.

3. PCTER.

4. Priorités fixes.

5. Algorithmes multi files à trois files (4 versions).

    a) Avec ou sans migration.

    b) Les deux premières files sont ordonnancées par des tourniquets de quantum q1 et q2 et :
        i. La troisième file est ordonnancée par un tourniquet de quantum q3 (Version 1).
        ii. La troisième file est ordonnancée par FIFO (Version 2).

Dans le menu déroulant les exercices seront affichés comme ceci :

-Tourniquet.

-FIFO.

-PCTER.

-Priorités fixes.

-Algorithme multi files FIFO sans migration.

-Algorithme multi files FIFO avec migration.

-Algorithme multi files TOURNIQUET sans migration.

-Algorithme multi files TOURNIQUET avec migration.

Pour chaque exercice les objectifs, les données d'entrées et les réponses attendues sont identiques :

Données : la première donnée entrée est l'algorithme d'ordonnancement.

Ensuite le système de processus est entré. Les informations entrées sont :

-Le nombre de processus.
-Pour chaque processus :
- Sa date d'arrivée.
- Sa durée.
- Dans le cas d'algorithme à priorité, sa priorité qui est un entier. Plus l'entier est petit, plus la priorité est forte.

- Si c'est un tourniquet le quantum q.
- Si c'est un multi files tourniquet, q1, q2 et q3.
- Si c'est un multi files FIFO q1 et q2.

-Réponses attendues :

- Un diagramme de Gantt vide et de durée égale à la somme des durées des processus est affiché et l'utilisateur doit le remplir, donc il remplit les cases en indiquant quel processus doit s'exécuter instant par instant.
- Calcul du temps de réponse, sous la forme d'une fraction numérateur sur dénominateur.

-Contraintes :

-Le nombre maximal de processus est compris entre 4 et 10.
-La durée d'un processus est entre 1 et 8.
-La valeur maximale du quantum est entre 1 et 4.
-Les valeurs des priorités sont comprises entre 1 et 4.

-Contraintes supplémentaires dans le cas aléatoire.

-Les dates d'arrivées des processus doivent être en ordre croissant.
-La date d'arrivée d'un processus doit être inférieure ou égale à la somme des durées de processus déjà arrivés avant lui.

L'exemple d'interface ci-dessous est l'unique interface pour les différents exercices

Explication fonctionnement de cette interface :

Les boutons valider et nouveau sont bloqués puisque l'utilisateur peut seulement dans un premier temps choisir le type d'ordonnancement voulu.
L'étape suivante permet d'entrer le nombre de processus et, si l'algorithme choisi est un tourniquet, le quantum si c'est la version manuelle, sinon les valeurs sont générées aléatoirement. On clique sur Go et le tableau s'affiche, il sera rempli si c'est la version aléatoire et sinon l'utilisateur le remplit.
On clique sur le bouton afficher case et les cases s'affichent automatiquement si c'est la version aléatoire et si c'est la version manuelle la syntaxe est vérifiée et si elle est correcte les cases s'afficheront, sinon un message d'erreur indiquera que la saisie est incorrecte.
L'utilisateur maintenant remplit les cases en indiquant quel processus s'exécute à chaque instant et puis valide sa réponse. Si c'est la bonne, il passe au temps moyen de réponse. Il dispose de 3 tentatives pour saisir les bonnes réponses.
Une fois les saisies réalisées, le bouton valider sera disponible pour que l'utilisateur valide son résultat et le bouton nouveau est activé après la validation, pour permettre à l'utilisateur de relancer l'exercice avec de nouvelles données.

<u>En cas d'erreur:</u>

En cas d'erreur que ce soit dans les données de départ ou dans les réponses, il est important de préciser où se situe précisément l'erreur via les messages box et non simplement « erreur ».

Pour le processus mettre en rouge la case où il y a une mauvaise réponse permet de simplifier les choses comme dans l'exercice des tableaux.
Après 3 tentatives incorrectes, le bouton valider est bloqué et le nouveau est débloqué.

Rappels :

- Temps de réponse pour un processus = Date de fin - Date d'arrivée.

- Temps moyen de réponse = somme des temps de réponse de chaque processus divisé par nombre de processus.

1) Tourniquet

Contraintes :

-Voir les contraintes générales

Rappels méthodologiques :

- Le processus sélectionné est celui en tête de liste et chaque nouveau processus est ajouté en fin de liste.
- Un processus en cours s'exécute pendant un temps (q) et se met en fin de file s'il n'a pas fini.

2) FIFO

Rappels méthodologiques :

- FIFO = First-In, First-Out, le premier processus arrivé est le premier à s'exécuter.

3) PCTER

Rappels méthodologiques :

- Le processus sélectionné est celui qui a le plus court temps d'exécution restant.

4) Priorités fixes

Rappels méthodologiques :

- Comparaison des processus pour trouver celui avec la plus forte priorité.
- Respecter l'ordre de priorité : si un nouveau processus a une priorité plus forte que le processus en cours d'exécution celui-ci est interrompu et est remplacé par ce nouveau processus.
- Lorsque deux processus ont la même priorité, la règle du FIFO s'applique.

5) Algorithmes multi files à trois files

Rappels méthodologiques pour les 2 versions :

Quand une file est ordonnancée par un quantum q, elle applique son algorithme d'ordonnancement pendant q unités de temps, sauf si elle se retrouve vide avant l'expiration du quantum dans ce cas la file suivante prend le relai.

Quand une file est interrompue, elle reprend là où elle en était, donc si un processus n'avait pas fini son quantum, il le termine.

Un processus dans une file donnée ne peut s'exécuter que si les files plus prioritaires sont vides.

a) <u>Version 1 Les 2 premières files sont en quantum q1 et q2, la 3 -ème files ordonnancée par un tourniquet de quantum q3</u>

Rappels méthodologiques :

Le principe d'une multi file avec migration est que les processus commencent tous dans la première file et puis passent dans la 2ème et finalement dans la 3ème sauf si le processus s'est achevé dans la 1ère ou à 2ème file. La file 1 a la plus haute priorité puis la seconde et la 3ème à la priorité la plus faible.

Le déroulement avec migration se fait de cette façon, le processus fait un tour dans la file 1 de quantum 1 puis passe dans la 2ème exécute un tour de quantum 2 puis passe dans la 3ème file de quantum 3, si le processus ne se termine pas du premier tour dans la 3ème file il se remet à la fin comme un tourniquet normal jusqu'à exécution complète.

Le principe d'une multi file sans migration est que chaque processus qui arrive possède une date d'arrivée, une durée et un niveau de priorité 1, 2 ou 3 qui indique dans quelle file il s'insère. Il reste ensuite toujours dans cette file. La file 1 a la plus haute priorité puis la seconde et la 3ème à la priorité la plus faible.

+ rappel tourniquet écrit précédemment

b) <u>Version 2 Les 2 premières files sont en quantum q1 et q2, la 3 -ème files ordonnancée par FIFO</u>

Rappels méthodologiques :

Même principe que tourniquet mais la 3ème file est FIFO.

- FIFO = First-In, First-Out, le premier processus arrivé est le premier à s'exécuter.

CORRECTIF

Après la complétion du thème “ordonnancement” plusieurs modifications ont été apportées à l’application. Ces modifications ont été réalisées suite à l'accord du client et ces dernières sont:

Au niveau de l’interface:

→ Suite à la sélection du thème ordonnancement l’exercice ne s’ouvre plus avec un menu déroulant mais une nouvelle page contenant les différents exercices sera affichée et l’utilisateur choisira l’exercice souhaité en manuel ou aléatoire.

Dans l’exercice, le menu déroulant qui était présent avant a été remplacé par un bouton confirmer pour permettre à l'utilisateur de confirmer son choix. Sinon il peut cliquer sur le bouton quitter pour retourner au choix entre aléatoire et manuel ensuite il pourra cliquer sur le bouton retour pour retourner à l'interface proposant tous les exercices.

Dans l’ancienne version si l’algorithme choisi ne contenait pas de quantum la case était bloquée mais maintenant la case n’est pas présente dans ces types d’algorithme.