Prog UNIX : Outils du programmeur

TP n°4

L'objectif de ce TP est d'identifier la meilleure méthode pour calculer les nombres binomiaux. Il s'agit donc d'un problème d'algorithmique et de complexité. Pour vous faciliter la tâche, un projet non terminé et donc résolvant partiellement le problème est disponible à l'adresse suivante :

http://www-igm.univ-mlv.fr/~borie/cours/prog_sys/projet_binomial.tar.gz

On rappelle que le nombre binomial de n et p , noté $\binom{n}{p}$ est défini par :

$$\binom{n}{p} = \frac{n!}{p!(n-p)!}$$

Pour tout $n \geq 1$ et $p \geq 1$, on a aussi la relation :

$$\binom{n}{p} = \binom{n-1}{p} + \binom{n-1}{p-1}$$

Les conditions aux bords du triangle de Pascal sont :

$$\forall n \in \mathbb{N}, \binom{n}{n} = \binom{0}{0} = 1$$

Pour ce TP, il vous sera demandé de rendre un compte rendu de TP décrivant la manière dont vous avez résolu chaque exercice. Ce compte rendu devra être un fichier texte nommé TP4_NOM_PRENOM.txt où NOM est votre nom de famille et PRENOM votre prénom.

Exercice 1 Makefile pour ce projet

Une fois le code téléchargé, survoler les sources afin d'écrire un Makefile pour compiler ce projet. Faire en sorte que l'exécutable final porte le nom binomial. Ajouter votre Makefile dans votre fichier de commandes.

Exercice 2 Erreur de Segmentation

Maintenant que le projet compile, exécuter le programme binomial avec les arguments 12 et 6. Que se passe-t-il ? Utiliser le débogueur gdb pour localiser la ligne de code générant l'erreur et corriger cette ligne. Expliquer la correction effectuée dans votre fichier de commande.

Exercice 3 Gestion de la mémoire

Utiliser l'utilitaire valgrind pour faire un diagnostic mémoire du programme. Repérer la fuite de mémoire le cas échéant et corriger le code pour obtenir le message no leaks are possible. Expliquer la correction à faire dans votre fichier de commandes.

Exercice 4 Profilage des trois méthodes de calcul

Utiliser l'utilitaire gprof pour profiler le projet. Regarder le nombre d'appels pour chacune des trois méthodes de calculs avec les données de type $2n, n$ où n est entier. Il s'agit donc de calculer et profiler les calculs de $\binom{4}{2}, \binom{6}{3}, \binom{8}{4}, \dots$

Quel est selon vous la meilleure méthode et pourquoi ? Quels sont les avantages et les inconvénients de chaque méthode ? Soyez précis et exhaustifs !

Exercice 5 (*) Calcul de complexité

Il s'agit maintenant de tenter de montrer de manière un peu plus mathématique les impressions données par les résultats du profilage.

Quel est la complexité algorithmique d'une multiplication par rapport à une addition ? Quel est, finalement, la méthode la moins coûteuse pour calculer $\binom{2n}{n}$ pour des grosses valeurs de n ?

Il n'est pas attendu ici une véritable démonstration mathématique mais un maximum d'indices et d'arguments donnés par les outils de programmation. Une méthode propre consiste à utiliser les fonctionnalités de `time.h` avec la commande `clock`. (le manuel `man` et internet peuvent vous aider).

Exercice 7 (*) Graphe de complexité**

En utilisant l'utilitaire `gnuplot` (aider vous d'internet), tenter de générer un graphe proposant trois courbes. En abscisse placer un entier n et chaque courbe donne pour l'ordonnée le temps de calcul en seconde de $\binom{2n}{n}$.

Exercice 8 Finalisation du rendu

Voici le cahier des charges pour ce quatrième TP :

- votre travail devra être rendu en pièce attachée d'un mail envoyé à une adresse communiquée par votre chargé de TP,
- vous ne devrez envoyer qu'une seule pièce jointe non compressée dont le nom devra être : `TP4_NOM_PRENOM.txt` où `NOM` est votre nom de famille et `PRENOM` votre prénom,
- le sujet de votre mail devra être : `[DUT1 info][TP4 prog_sys] NOM PRENOM` où `NOM` devra être remplacé par votre nom de famille et `PRENOM` par votre prénom,
- la qualité du fichier rendu sera fortement prise en compte par le correcteur.