Université de Strasbourg
Algorithmique et Programmation 1 Feuille de TP

Introduction à Objective CAML sous Linux

Dans ce premier TP, vous allez apprendre à utiliser le langage de programmation Objective CAML. Pour ce faire, nous allons utiliser le système d'exploitation Linux qui offre un meilleur environnement de programmation que MS-Windows. Néanmoins, Objective CAML est également utilisable sous Windows, ou bien sur un Macintosh (voir annexes).

De plus, l'interpréteur Objective CAML est un logiciel libre, ce qui fait que vous pouvez en obtenir une copie gratuitement pour l'installer chez vous (voir annexes).

I) Le système d'exploitation Linux

Le système d'exploitation est le logiciel de base d'un ordinateur, c'est lui qui fournit les primitives d'accès aux composants matériels de la machine (microprocesseur, mémoire, écran, clavier, souris, disque dur, lecteur de DVD, imprimante, accès réseau, etc.). Les différentes applications que l'on utilise sur un ordinateur font appel au système d'exploitation pour échanger des informations avec ces composants de manière indépendante de leur caractéristiques techniques exactes (marque, modèle, etc.).

Sur les ordinateurs de type PC, les systèmes d'exploitation les plus répandus sont les différentes variantes de Windows développées par la société Microsoft. Mais ce n'est pas le seul, il y a en particulier le système d'exploitation Linux, développé par un ensemble de programmeurs répartis dans le monde entier, et diffusé sous une « licence[1] » qui permet à chacun de copier, modifier et redistribuer Linux comme il le désire. Ce système d'exploitation fait ainsi partie de la famille des *logiciels libres*, pour lesquels on ne paye que le support (CD-ROM en général), la documentation, ou l'assistance technique quand on décide de les acheter ; il n'y a aucune obligation de payer des droits d'auteurs ou des « licences » d'utilisation.

Linux est un système d'exploitation de la famille UNIX, et la plupart des aspects présentés ici sont en fait valables pour tous les UNIX.

Pour travailler sous UNIX il est nécessaire au préalable de se faire identifier par le système comme un utilisateur répertorié. C'est pourquoi il vous faudra commencer par donner votre nom d'utilisateur (*login name*) et votre mot de passe (*passwd*) avant de commencer à travailler sur l'ordinateur.

II) Les arborescences des répertoires et des fichiers

Sous UNIX, les fichiers sont regroupés par paquets, formant des répertoires et sous-répertoires, et globalement une arborescence de fichiers, de manière analogue à Windows.

Les noms de fichiers sont limités à 256 caractères. Il n'est pas obligatoire d'utiliser une extension du genre ".pdf" pour indiquer le type du fichier. Néanmoins, on continuera à utiliser cette convention pratique. Il est recommandé d'éviter les lettres accentuées et espace dans les noms de fichier. **En particulier, les programmes CAML devront toujours être écrits dans des fichiers se terminant par `.ml`.**

[1] Cette licence est la « General Public License » ou GPL en abrégé. Voir `http ://www.gnu.org/home.fr.html` pour plus d'informations.

a) Le répertoire HOME

Vos fichiers personnels, se trouvant dans votre répertoire HOME, sont stockés de manière permanente sur une machine du réseau local. Vous retrouverez vos fichiers d'une session à l'autre, et ceci même si vous vous installez sur un autre ordinateur du réseau.

Pour manipuler les fichiers (copie, renommage, effacement, déplacement, etc.), vous disposez d'un gestionnaire de fichiers, obtenu en cliquant sur l'icône HOME (petite maison).

Vous pouvez également faire ces manipulations par des commandes clavier qu'il faut lancer dans un *interpréteur de commandes shell* (encore appelé *terminal*, ou *console*). Ces commandes sont décrites dans la fiche distribuée séparement.

Exercice 1

1. Cliquer sur l'icône HOME du bureau afin de voir le contenu de ce répertoire dans un gestionnaire de fichiers.
2. Créer un sous-répertoire `temp`.
3. Ouvrir un deuxième gestionnaire de fichiers en cliquant à nouveau sur HOME.
4. Depuis le web, sauvegarder le fichier `tp-intro.ml`.
5. À l'aide de la souris, copier le fichier `tp-intro.ml` dans le répertoire `temp`.
6. Lancer un terminal, et faire afficher la liste détaillée des fichiers du répertoire HOME.
7. À l'aide d'une commande clavier, effacer le fichier `tp-intro.ml` et le répertoire `temp`.
8. À l'aide d'une commande clavier, créer un répertoire `AlgoProg1` puis copier le fichier `tp-intro.ml` dans ce répertoire. Utiliser la touche TAB de complétion pour éviter de tout taper.

Exercice 2

1. Vous allez maintenant éditer (c-à-d. modifier) le fichier `tp-intro.ml` (qui contient un programme) à l'aide d'un lociciel (de la famille des éditeurs de texte) qui s'appelle *kate*. Pour cela, exécutez la commande shell "`kate tp-intro.ml &`" dans un terminal.
2. Le programme contenu dans `tp-intro.ml` doit être interprété en Ocaml. Pour cela, exécutez *dans le terminal intégré à kate* la commande shell `ocaml` (ou bien `ledit ocaml` pour plus de souplesse). Testez l'interpréteur avec quelques commandes simples.
3. Devinez ce que vous allez obtenir en évaluant les expressions données dans le fichier. Evaluez les définitions avec Ocaml pour vérifier vos réponses. Corrigez les erreurs. L'outil "transmettre à la console" peut vous simplifier la tâche.

Exercice 3

Soit le polynôme $P(x) = 2 + 3 \times x + 4 \times x^2 + x^3 + 2 \times x^4$. On cherche à coder la fonction `poly` dont le prototype est `int -> int` et qui calcule $P(x)$ pour un paramètre x entier.

1. Proposer un premier codage direct de la formule. Combien de multiplications sont nécessaires ?
2. Proposer un deuxième codage qui commence par calculer x^2, x^3 et x^4 avant d'évaluer le polynôme P en x. Combien de multiplications sont nécessaires ?

3. En utilisant la distributivité, on peut écrire un polynôme $P(x) = a_0 + a_1x + ... + a_nx^n$ sous la forme $P(x) = a_0+x(a_1+x(a_2+...x(a_{n-1}+xa_n)...))$. Utiliser cette propriété pour proposer une autre méthode de calcul de la valeur du polynôme P. Combien de multiplications sont nécessaires ? Cette technique est connue sous le nom de méthode de Hörner.

Exercice 4

Etant données les expressions suivantes :

```
let a = 6 ;;
let f x = x + a ;;
f 4 ;;
let a = 2 ;;
f 4 ;;
```

Qu'obtient-on après l'évaluation de la dernière ligne ?

Exercice 5

1. Ecrire des fonctions `double` et `carre` qui retournent respectivement le double et le carré de leur argument entier.
2. Ecrire une fonction qui compose les deux fonctions passées en argument. Quel sera le type de cette fonction ? Appliquer cette fonction aux fonctions `double` et `carre`.

III) Annexes

a) Obtenir la distribution ocaml

Si vous voulez travailler avec OCaml chez vous : sous linux, la plupart des distributions contiennent OCaml ; et sous Windows ou MacOS X vous pouvez charger la distribution du serveur de l'INRIA : `http://caml.inria.fr/download.en.html`.

b) Conseil pour la création d'un mot de passe

Le mot de passe que vous choisissez pour protéger votre compte doit être suffisament complexe pour ne pas être deviné par n'importe qui, par contre vous devez pouvoir vous en souvenir facilement[2].

Voici quelques règles pour la création d'un bon mot de passe :

- votre mot de passe doit faire au moins 6 caractères,
- votre mot de passe doit contenir des caractères autres que des lettres (chiffres et symboles de ponctuation...),
- votre mot de passe ne doit pas être directement un mot ou un nom propre...

Pour créer un bon mot de passe, on peut par exemple choisir une petite phrase facile à retenir (comme "ocaml for linux"), on remplace ensuite certaines lettres par des chiffres de forme similaire (ici on remplacera le o de "ocaml" par un 0), on peut egalement remplacer certains mot d'une seule syllabe par un chiffre de prononciation similaire (dans notre exemple le "for" peut se remplacer par le chiffre 4, qui se prononce *four* en anglais), enfin il faut réduire la phrase à 8 caractères en coupant certains mots et en

[2]Il est impossible de récupérer directement un mot de passe, si vous oubliez le votre, il faudra en créer un nouveau. Cette opération ne peut se faire qu'avec l'ancien ou, le cas échéant, ne peut être effectué que par un administrateur système.

enlevant les espaces (dans notre exemple on obtient “0caml4li”) et on rajoute quelques majuscules (ce qui nous donne “0cAml4Li”). Le mot de passe obtenu ainsi est en général facile à retenir et suffisament complexe pour ne pas être découvert par un programme d’analyse de mot de passe.