Licence d'informatique
Module de C/Unix

Projet Unix

m2h — Contrôle de TP

Philippe MARQUET

Novembre 2000

1 Rendre votre projet et votre contrôle

Vous devez rendre à votre enseignant de TD/TP, par email :

- Votre projet constitué des fichiers suivants :
  - un fichier `Readme` comportant votre nom, celui de votre binôme, expliquant les fonctionnalités traitées dans votre implantation et celles qui ne le sont pas, donnant quelques détails que vous jugerez utiles sur votre projet (pages de manuels testées avec succès par votre commande...) ;
  - le fichier Bourne-shell de votre commande.
- Votre contrôle de TP sous cette même forme d'un fichier `Readme` et d'un fichier Bourne-shell. Tous les étudiants d'un groupe n'ayant pas le même sujet à traiter, vous identifierez clairement le sujet traité.

Les adresses mails des enseignants sont les suivantes :

Groupe 1 –	`Jean-Luc.Levaire@lifl.fr`
Groupe 2 –	`Philippe.Marquet@lifl.fr`
Groupe 3 –	`Aurelien.Lemay@lifl.fr`
Groupe 4 –	`Julien.Cartigny@lifl.fr`

2 Pages de manuel en anglais

Il vous est demandé de manipuler les sources de pages de manuel. On ne considère que la version originelle en anglais des pages. Il est possible que des traductions françaises de pages soient disponibles sur la machine sur laquelle vous travaillez. Pour éviter d'utiliser ces pages, il vous faut supprimer les définitions des variables d'environnement `$LANG` et `$LANGUAGE`.

3 Mots-clés dans les pages de manuel

La section `KEYWORDS` d'une page de manuel liste les mots-clés se rapportant à la commande décrite dans la page. Cette section est une liste de mots-clés séparés par des virgules. Voyez par exemple les pages de `tclsh(1),wish(1),tixwish(1).`

Nous allons développer la commande `keyman` qui affiche le nom des commandes dont la page de manuel contient un mot-clé donné. les pages de manuel scrutées sont celles du répertoire donné en paramètre, `/usr/man` par défaut. La syntaxe de la commande est la suivante :

`keyman` [*`directory`*] *`keyword`*

Exercice 1
Développez la commande `keyman` en Bourne-shell.

Id : m2h-xam.tex,v 1.2 2000/11/16 15 :56 :22 marquet Exp

4 Entête d'une bibliothèque

La section `SYNOPSIS` de la page de manuel d'une bibliothèque peut comporter des lignes d'inclusion de fichiers (directives `cpp` de la forme `#include`). Voyez par exemple la page de manuel de `open(2)`.

Nous allons développer une commande `header` qui, pour une page de manuel donnée (sous forme d'un fichier `troff` éventuellement compressé ou sous forme d'un nom de commande éventuellement précédé d'une section), affiche les directives `cpp` d'inclusion. La syntaxe de la commande est la suivante :

`header` *file*
| `header` [*section*]*command*

Exercice 2
Développez la commande `header` en Bourne-shell.

5 Prototype d'une fonction de bibliothèque

La section `SYNOPSIS` de la page de manuel d'une fonction de bibliothèque comporte une (ou plusieurs) ligne qui donne le prototype de la fonction. Voyez par exemple la page de manuel de `open(2)`.

Nous allons développer une commande `proto` qui, pour une page de manuel donnée (sous forme d'un nom de commande éventuellement précédé d'une section), affiche le prototype de la fonction donnée. La syntaxe de la commande est la suivante :

`proto` [*section*]*command*

Exercice 3
Développez la commande `proto` en Bourne-shell.

6 Rapporter un bug par email

Les sections `AUTHORS` et/ou `REPORTING BUGS` d'une page de manuel peuvent contenir l'adresse email de l'auteur ou une adresse email à laquelle rapporter un bug. Ces adresses email sont présentées entre < et >. Voyez par exemple les pages de manuel de `tail(1)` ou de `cdecl(1)`.

Nous allons développer une commande `mailbug` qui, pour une page de manuel donnée (sous forme d'un nom de commande éventuellement précédé d'une section), affiche les adresses email listées dans ces sections `AUTHORS` et/ou `REPORTING BUGS`. La syntaxe de la commande est la suivante :

`mailbug` [*section*]*command*

Exercice 4
Développez la commande `mailbug` en Bourne-shell.