Bailhache Pierre
Ladoucette Arnaud
Ly Trong-Quy

Groupe A2IN501
Samedi 8 Décembre 2012

Projet de Système

Introduction

Nous avons donc réalisé un programme qui copie le fonctionnement de la commande UNIX tar. Pour cela nous devions mettre en place un système d'archivage des dossiers et des fichiers. Pour réaliset ce système nous avons donc utilisé une librairie xml, la libxml2 présente de bases sur la plupart des distributions linux. Nous avons choisi le XML plutôt qu'un système de header de fichier car l'interface de sauvegarde et d'extraction s'en retrouver plutôt simplifié mais demander aussi de l'adaptation. De plus la gestion des répertoires est presque native avec le système à base de XML.

Ce dossier va présenter les trois fonctions principales du programme, la création, l'extraction et le système de données pour la sauvegarde de l'archive. Il présente aussi la gestion des options du programme.

Creation d'une archive

Récupération des informations

Dans notre programme d'archivage, nous conservons la date de création et les droits. Pour simplifier cette sauvegarde nous avons créer une structure regroupant des informations essentiel d'un fichier :

- nom : utiliser pour l'identifier et pour sa re-création.
- date de création sous forme d'un time_t.
- droits : sous forme de mode_t, dit aussi sur le type de fichier.
- taille: utiliser pour récupéré les données.

Comme la date, les droits et la taille peuvent se convertir sous forme d'entier, on a utiliser les fonctions *htonl()* et *ntohl()* pour assurer une compatibilité avec d'autre processeurs.

Parcours des fichiers et dossiers

la lecture des fichiers à archiver se fait d'une manière récursive pour parcourir les dossiers passés en paramètre. Nous avons choisi de faire le parcours des dossiers en faisant des

changement de répertoire via la fonction *chdir()* car elle évite de gérer des chemins de fichier avec des "/".
Avec la fonction *scandir()* on récupère un tableau contenant tous les fichiers d'un répertoire. cette fonction à l'avantage de proposer un tri alphabétique et donc ignorer les dossiers "." et ".." qui dérangent l'appel récursif lors du parcours total.

Enregistrement du fichier

L'archive s'enregistre après que le XML soit complet, avec l'option *-f* il s'enregistre dans un chemin passé en paramètre. Si l'utilisateur n'indique pas de fichier, par défaut le fichier de sortie sera sous le nom de "*Archive_yyyy-mm-dd_hh-mm-ss.tarx*".

Extraction d'une archive

Le programme effectue en quelque sorte le chemin inverse de la création : il parcourt tous les fichiers à la racine de l'archive et récupère les information avec la structure *file_info*. Si le fichier est un dossier, alors on créer d'abord le dossier (en restaurant correctement les droit et la date de création) puis on change le répertoire courant (*chdir*) pour ensuite exploré tous ses fichiers dans l'archive.

L'utilisateur peut choisir de mettre un chemin d'extraction pour l'archive

Présentation de la partie XML

Structure du document XML

Nous avons donc choisi d'utiliser du XML pour l'archivage des fichiers. Il y deux types d'entités à archiver : les dossiers et les fichiers. Tout le document est organisé en un grand arbre constitué de *xmlNode*. Les dossiers et les fichiers sont donc des *nodes*, les premiers possèdent des fils et les deuxièmes n'ont pas d'enfants dans l'arbre.

Il y a une *Node* racine qui possède le nom de l'archive s'il existe et *"./"* s'il existe pas.

Chaque *node* possède un nom l'identifiant dans le fichier ainsi qu'un ensemble d'attributs. Ces attributs sont :

"Time" : Correspond au *create_time* de la structure *file_info*, heure de création du fichier.
"Mode" : Correspond au *mode* de la structure *file_info*, c'est les droits.
"Size" : Correspond au *size* de la structure *file_info*, la taille du fichier ou du répertoire.
"Data" : Uniquement réservé aux fichiers, c'est l'attribut qui contient les données du fichier archivé.

Tous les attribut sont enregistrés en *xmlChar** puis sont reconvertis lors de l'extraction des fichiers.

Création et extraction du XML

Création

La création du XML se fait lors du parcours des fichiers et des répertoires. Il y a deux fonctions importantes, *AddFolder()* et *AddFile()*. Les deux fonctions prennent en paramètre une *file_info*, un *char** correspondant au nom, un *xmlNodePtr* pointant sur le répertoire parent de l'élément à ajouter. Les deux fonctions marchent de manière similaire excepté le fait que *AddFile()* prend un paramètre supplémentaire correspondant aux données du fichier. *AddFolder()* retourne un pointeur sur le répertoire qu'on vient de créer pour pouvoir ajouter des fichiers et des répertoires.

Extraction

Pour l'extraction, on récupère la liste des fichiers et des répertoires via *tar_root_files()* à la racine. Ensuite la fonction *tar_folder_files()* permet de récupérer la liste à partir d'un répertoire défini par son nom et le nom de son répertoire père. La récupération des données s'effectue ensuite par la fonction *get_tar_data_file()* en précisant le nom du répertoire contenant le fichier. Pour récupérer les pointeurs sur les *nodes* représentant les fichiers et les répertoires, nous utilisons *XPath*. En utilisant une expression du type *//nomDuPere/nomDuFichier* on retrouve le fichier dans l'arbre sans avoir à faire un parcours complexe et donc avec un temps de calcul plus court.

Gestion des options

Certaines options demandes des listes de fichiers en arguments comme l'option -c. Pour récupérer correctement cette liste, on impose dans l'utilisation de notre programme de mettre ces listes de fichiers en dernière position. En effet il est plus facile de les récupérer via *&argv[optind-1]* où *optind* est la position du premier fichier qui suit l'option voulu. De plus cela simplifie la prise en compte des autres options puis qu'ils seront placé avant.

Conclusion

Notre programme rempli une grande partie des fonctionnalités demandés. La création est complètement fonctionnel avec création du fichier d'archive. L'ajout de fichier et la suppression dans une archive sont fonctionnels aussi. Nous avons implémenté l'affichage de l'archive avec l'option *-t*. L mode *verbose* est prêt mais non fonctionnel suite à un manque d'organisation.

La principale fonction qui ne fonctionne pas est l'extraction des fichiers de l'archive via *-x.* Ceci est du à un problème de segmentation qui provient d'une mauvaise conversion des données stockées dans la structure *file_info* et leur lecture.