

Université Paris Diderot
Master 1 Informatique
2013-2014

PROJET **PROGRAMMATION SYSTEME**

Alejandro CASTELLANOS
Meryam JAMAL
Erica HUAM

Usage programme :

```
make  
./projet fichier.dzb
```

Lecture du Dazibao:

Fichiers : load_daz.c & projet.c

verif_entete.c : lecture et affichage de l'entête d'un dazibao

Fonction utilisée :

- int verif_entete(unsigned char*tmp)
 tmp : correspond au premier 4 bits

load_daz.c : lecture et affichage des TLVs d'un dazibao

Fonctions utilisées :

- int calcul_length (unsigned char *tmp,int i) : fonction calculant la taille d'un tlv
- void affiche_date(time_t date) : fonction qui convertit et affiche la date
- lect_tlv(unsigned char *tmp, int i,int taille) :
 tmp : contient tous les caractères du fichier dazibao

Procédure :

On copie tous les caractères du dazibao dans un buffer .

Ajout dans le Dazibao :

Fichiers : ajout.c & projet.c/type_menu

ajout.c : ajoute des TLV à la fin d'un dazibao

Fonctions utilisées :

```
add_pad_n(int fd, int taille, int indice, int type)  
int add_tlv_txt1(dazibao *dazchargee, int type, int fd, int taille, int dated)  
add_tlv_picture(dazibao *dazchargee, int type, int fd_daz, int taille, int  
dated)  
add_tlv_compound(dazibao *dazchargee, int type, int fd, int taille, int  
flag)
```

Procédure :

- Poser le verrou pour écrire dans le dazibao
- Écrire les tlvs copier dans le buffer
- Enlever le verrou

Suppression dans le Dazibao :

Fichier : projet.c

Fonctions utilisées :

```
int del(char * path, int delete_index)  
delete_index : correspond au classement du tlv dans le dazibao
```

Procédure :

- Poser le verrou
- Parcourir le fichier pour trouver le tlv correspondant. On remplace alors le type du tlv par PadN, et value par des 0 binaires .
- Enlever le verrou

Compaction :

Fichier : compaction.c

Fonctions utilisées :

```
int decale(char * path,int curs, int nb_bytes_a_dec, int size)
void compact_tlv(char * path, unsigned char * contenu, int daz_size)
int compact(char * path)
```

Procédure :

- Poser le verrou
- Parcourir le fichier
 - si Pad1 : décale tous les caractères d'une case
 - si PadN : se positionne à la fin du PadN , on décale les caractères suivant de la taille (length) du PadN
- pour trouver le tlv correspondant. On remplace alors le type du tlv par PadN, et value par des 0 binaires.
- enlever le verrou

=> Cette méthode permet d'optimiser la mémoire

Sockets :

Fichier : serveur.c & clients.c

serveur.c : met en place le serveur

Fonctions utilisées :

```
void write_date(int fd, int ligne, time_t date) : Écrit la date dans le fichier
pointé par le file descriptor fd à la ligne indiquée. Format de la date :
YYYYMMDDHHMMSS
int memdate(int ligne, time_t date) : Compare date à la date stockée à la
ligne indiquée du fichier $HOME/notif.txt
```

`char ** init(int nb_daz, char ** argv)` : Initialise notif.txt qui contient une date de dernière modif par fichier .dzb. Les dates sont triées dans le même ordre d'apparition que les dazibaos de la liste `daz_paths`
`void notif(int * clients, char * path)` : Envoi la notification de la modification du dazibao pointé par `path` à tous les clients connectés.

Procédure :

- Initialise les fichier notif.txt situé dans le répertoire HOME avec les dates de dernières modifications des dazibaos passés en paramètre
- Met en place le socket serveur pour lui permettre d'accepter des connexions
- Boucle sur un accept en créant deux processus : le père gère les nouvelles connexions tandis que le fils s'occupe des modifications des dazibaos.

client.c : met en place les clients

Fonction utilisée :

`main(int argc, char ** argv)`

Procédure :

- Met en place un socket client pour lui permettre de se connecter au serveur
- Boucle sur un read en attente d'un message du serveur indiquant si un dazibao a été modifié. Si c'est le cas, affiche son path sur le terminal.