Instituto Superior Técnico

Electrical and Computer Engineering

Systems Programming

2017/2018 2nd Semester

Project Assignment

Distributed Clipboard

Authors

André Agostinho	84001	andre.f.agostinho@tecnico.ulisboa.pt
João Pinheiro	84086	joao.castro.pinheiro@tecnico.ulisboa.pt

Contents

1 Introduction **2**

2 Architecture **2**
2.1 Source Tree 2
2.2 Communication Protocol 2
2.3 Connection Handling 3

3 Threads **3**
3.1 Shared Data 3
3.2 Synchronization 3

4 Replication **3**

5 Error Handling **3**

1 Introduction

The goal of this project was to implement a distributed clipboard and an application interface to it. Applications can copy (write) and paste (read) from and to regions on a local clipboard, which can be connected to another remote clipboard so as to synchronize their regions. The resulting network is a tree, as a clipboard can only connect to one other (the parent), but it can itself have connections to various children clipboards. There is always one clipboard which did not start by being connected, and that is the root of the tree. This tree is shown in <le figure ref>

2 Architecture

2.1 Source Tree

```
/
├── apps/
│   ├── copy.c // Hello
│   ├── paste.c
│   ├── paste
│   └── copy
├── cb_common/
│   ├── cb_common.h
│   ├── cb_msg.h
│   └── cb_msg.c
├── library/
│   ├── libclipboard.h
│   └── libclipboard.c
└── clipboard/
    ├── app.h
    ├── conn.h
    ├── parent.h
    ├── child.h
    ├── child.c
    ├── conn.c
    ├── clipboard
    ├── utils.c
    ├── clipboard.h
    ├── parent.c
    ├── app.c
    ├── utils.h
    ├── region.h
    ├── region.c
    └── main.c
```

2.2 Communication Protocol

To communicate between local applications and different clipboards, a simple protocol for messages was implemented as

```
uint8_t command;
uint8_t region;
uint32_t data_size;
uint8_t *data;
```

in which `data_size` is the size of the data in the region that is sent immediately afterwards this header. This message header is marshalled into a buffer (of bytes) for sending into the network as

0	1	2	3	4	5
`command`	`region`	`data_size`			
`data...`					

All the functions related to communication are in `common/cb_msg.c`, as they are shared by the library application interface and the clipboard server itself.

2.3 Connection Handling

Connections are handled in a one connection per thread model. There are three types of connections: local application connections, children clipboard connections and the optional connection to a parent clipboard. A connection, which is inexorably linked to a thread, is represented by a structure (in conn.h) as

```
struct conn {
    pthread_t tid;
    int sockfd;
    pthread_mutex_t mutex; // Protects writes to sockfd if needed
    struct conn *next;
    struct conn *prev;
};
```

in which the pointers `next` and `prev` are used to make a doubly-linked list of connections, of which there are two: local applications and children connections (in the implementation dummy head nodes are also used).

3 Threads

3.1 Shared Data

3.2 Synchronization

4 Replication

5 Error Handling