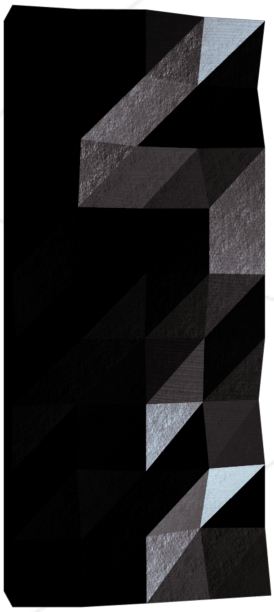

Parlons termcaps

ft_select

Résumé: Petite introduction aux termcaps.

Table des matières

Chapitre I

Les termcaps

Le projet `ft_select` fait une utilisation intensive de la bibliotèque `termcaps`. Mais qu'est ce que les `termcaps` ? `Termcap` désigne les "capacités" de votre terminal (deplacer le curseur, effacer une ligne, etc). C'est un peu les "tours" que votre terminal sait faire. Le truc, c'est que depuis l'invention des terminaux (a long time ago in a galaxy far, far away) on en trouve une fucktonne avec chacun leurs capacités, plus ou moins standardes, et surtout avec leur propre façon de les utiliser. Bref, c'etait un joyeux bazar et tout programme souhaitant être portable était moqué.

C'est alors que l'équivalent barbu des super heros s'est mis en quête d'organiser un peu ce foutoir. Tous les terminaux ont été inventoriés avec la liste de leur capacités afin d'avoir une base de données fiable des capacités proposées par l'ensemble des terminaux. Et ils ne sont pas arrêtés la ! Les barbus ont pris sur eux d'écrire une bibliothèque permettant de connaître les capacités disponibles pour un terminal donné et de les utiliser de manière UNIFIÉE ! Much awesome ! So work ! C'est ainsi qu'est née la bibliothèque `termcaps` que vous allez utiliser dans plsueiurs projets.

`Le texte précédent est une fiction qui n'a aucun rapport avec la réalité. Mais il permet une chronologie et une introduction aux termcaps abordable. Pour la véritable histoire, hop google. Toutefois, si l'un ou l'une d'entre vous a un talent certain pour le dessin, une mise en image de cette petite fiction sera appréciée et ajoutée à ce document.`

Une fois que vous avez trouvé votre terminal dans la base de données à l'aide de la fonction `tgetent(3)`, ses capacités se présentent sous la forme de flags booléens, d'entiers et de chaines de caractères. Il suffit alors d'utiliser la fonction associée au type de la capacité pour interroger la base de données : `tgetflag(3)`, `tgetnum(3)` et `tgetstr(3)`. Toutes les capacités existantes ont un code de deux caractères les représentant de manière unique. Par exemple, pour connaître le nombre de bras de votre terminal, vous ecririez :

```
int     nbarms;

nbarms = tgetnum("ar");
```

Ne cherchez pas à compiler cet exemple, car à ce jour aucun terminal avec des bras

n'a été répertorié et la fonction `tgetnum` renverrait peut-être `-1`, à moins que la capacité `"ar"` n'existe déja pour un truc sérieux.

Certaines capacités s'activent en écrivant une suite de caractères particuliers dans le terminal. Pour connaitre la chaine de caractères à écrire, il suffit d'interroger la base de données avec `tgetstr(3)` et d'écrire le résultat dans le terminal avec la fonction `tputs(3)`. Si votre terminal se tranforme en représentation d'art abstrait, c'est que vous avez raté un truc.

Il faut bien comprendre que certaines capacités de votre terminal s'utilisent en ecrivant une chaine de caractères dans votre terminal et d'autres en comparant une chaine reçue depuis votre clavier à la chaine d'une capacité. Par exemple les flèches directionnelles. Certaines touches ou combinaisons de touches ne sont pas représentées par un seul caractère, les flèches directionnelles sont dans ce cas. C'est très important de le savoir. Ainsi pour déplacer le curseur d'une colone vers la droite, il ne faut pas écrire la capacité "flèche droite" dans votre terminal. Il faut récupérer les caractères qui représentent "flèche droite" envoyés par votre clavier, puis les comparer à la chaine de la capacité "flèche droite" de votre terminal. Si les deux chaines sont identiques, vous pouvez écrire la capacité "déplacer le curseur d'une colone vers la droite" dans votre terminal. Vous voyez l'idée ?

Les termcaps, soyons francs, c'est pas très intuitif. C'est pourquoi lire leur doc officielle vous sera d'un grand secours. Prenez le temps de tester et de partager vos echecs.

Se pose aussi la question de la configuration de base de votre terminal. Jusqu'à maintenant, quand vous appuyez sur la touche `a` de votre clavier, vous ne vous étonnez pas de voir le caractère `'a'` s'afficher dans votre terminal. Vous ne vous étonnez pas non plus que votre terminal attende que vous ayez appuyé sur la touche `return` pour envoyer la ligne à votre `shell`. Pourtant ce sont là des comportements que vous ne souhaitez pas forcément selon votre programme. Il va donc falloir arranger ça en passant votre terminal en mode "raw" ou "non-canonique". Je laisse google vous expliquer pour moi. Sachez que les fonction `tcgetattr(3)` et `tcsetattr(3)` vous seront utiles. Etudiez le comportement de programmes comme `herrie` ou `mcabber` pour vous faire une idée de ce que je veux dire.

Quoiqu'il en soit, l'aventure `termcaps` commence par la lecture collective et approfondie de ces `mans` :

- termios
- termcap

Ces mans proposent une lecture longue et difficile ! Soyez intelligents ! Ne lisez pas ces mans comme des romans. Recherchez y des informations auquelles vous pouvez vous raccrocher et extrapolez à partir d'elles ! Ne les lisez pas seuls ! Regroupez-vous par petits groupes et lisez les phrases à haute voix, discutez-en et prenez des notes si nécessaire avant de passer à la phrase suivante. Une lecture solitaire de ces mans peut conduire à un sentiment profond d'insignifiance face au gigantisme de l'univers.