



# Proiect al modului C-Unix I

## ft\_select

*Sumar: Acest proiect are ca obiectiv sa scrieti un mic program cu ajutorul bibliotecii termcaps ce va permite selectarea unei optiuni valori dintr-o lista si returnarea ei in shell.*

# Cuprins

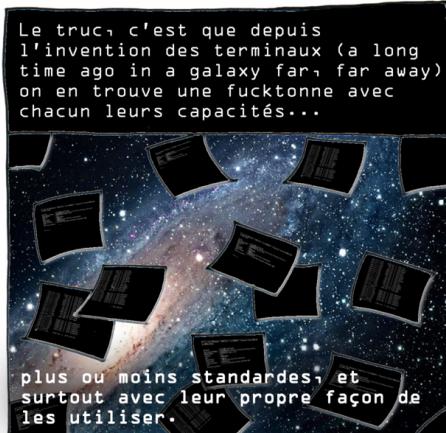
<b>I</b>	<b>Preambul</b>	<b>2</b>
<b>II</b>	<b>Subiect</b>	<b>4</b>
II.1	Parte obligatorie . . . . .	4
II.2	Bonus . . . . .	6
II.3	Livrare . . . . .	6
II.4	Consideratii tehnice . . . . .	8
II.5	Functii autorizate . . . . .	9
<b>III</b>	<b>Instructiuni</b>	<b>10</b>
<b>IV</b>	<b>Notare</b>	<b>12</b>

# Capitolul I

## Preambul

# LES TERMCAPS

Le projet ft\_select fait une utilisation intensive de la bibliothèque termcaps.  
Mais qu'est ce que les termcaps ?





# Capitolul II

## Subiect

Programe ca shell, `aptitude`, `top`, `tig`, `mcabber`, `dwarf` `fortress`, `zangband` sau `herrie` au toate cel putin doua lucruri in comun: aceste programe se executa intr-un terminal si au o interfata utilizator avansata in ciuda absentei unei interfete grafice de tip fereastra similara celor de la Microsoft Windows si OSX (ca si in folosirea Starbucks) cu care sunteți obisnuiti.

Realizati o interfata utilizator pentru un program care se poate executa intr-un terminal si necesita o munca de programator astfel ca aceasta sa controleze in mod "brut" un terminal, nu e mare lucru. Pentru a va convinge lansati comanda `cat` fara niciun argument si apasati pe taste sau pe combinatii ale acestora...

In aceasta actiune, in timp ce apasati pe tastele alfanumerice, nu se constata nimic deosebit. Daca insa veti apasa tastele sageti, `ESC` sau tastele `f n` de exemplu, vor fi afisate caractere arbitrar... Reluati apoi actiunea in shell-ul voastru. Sunteți obisnuiti ca la apasarea sagetii stanga sa se mute cursorul in acelasi sens cu o pozitie. De ce exista aceasta diferenta intre `cat` si shell?

Terminalul gestioneaza multe lucruri pentru voi fara ca sa va dati seama, ca de exemplu afisarea multor caractere propriu-zise (ex: linie multipla), precum si o serie de caractere de control.

Daca acum va intrebati cum puteti prelua controlul asupra terminalului, atunci acest proiect este destinat acestui scop. Veti invata sa configurati terminalul prin folosirea structurii `"struct termios"` si utilizarea capacitatiilor famoasei `"termcaps"`.

### II.1 Parte obligatorie

- Scrieti un program `"ft_select"` ce ia ca parametru o serie de argumente. Lista de argumente va fi afisata.
- Utilizatorul poate deci sa se deplaseze in lista de argumente cu ajutorul sagetilor (lista e circulara).
- Una sau mai multe optiuni pot fi selectate sau deselectate cu ajutorul tastei `spatiu`.

La fiecare selectie efectuata, cursorul trebuie automat pozitionat pe elementul urmator.

- Odata ce utilizatorul valideaza selectia facuta cu ajutorul tastei **return**, lista de optiuni trebuie returnata shell-ului. Optiunile trebuie sa se separeze de un spatiu. asta va permite deci utilizarea programului vostru in interiorul unui script shell (pentru a realiza un "set", de exemplu).
- Trebuie sa se poate scrie comenzile urmatoare:

```
$> set reponse = `ft_select choix1 choix2 choix3 choix4`  
$> more `ft_select *.c`  
$> rm `ft_select /*`  NB: Nu utilizati aceasta comanda DECAT daca  
                     sunteti SIGURI ca programul vostru ft_select  
                     functioneaza. Nu ne asumam nicio responsabilitate  
                     in cazul unui al programului vostru ft_select...
```

- Trebuie de asemenea sa gestionati redimensionarea ferestrei de catre utilizator. Lista trebuie afisata pe mai multe coloane daca dimensiunea ferestrei nu dispune de suficiente linii pentru a afisa totul pe o singura coloana. Daca un utilizator redimensioneaza fereastra in curs de utilizare, ansamblul optiunilor trebuie sa se rezizeze automat, iar optiunile selectate trebuie sa se pastreze. Cursorul trebuie sa fie pozitionat intr-un mod rezonabil dupa o redimensionare.
- Daca fereastra e prea mica pentru afisare (linii sau coloane insuficiente), atunci programul trebuie sa reacioneze in mod rezonabil pana cand dimensiunea este insuficienta. In niciun caz el nu trebuie sa se inchida. Din momentul in care fereastra va fi de dimensiuni suficiente, programul isi va relua functionarea normala.
- Daca un utilizator apasa tasta **ESC** programul nu va returna nimic shell-ului si se va termina normal.
- Daca utilizatorul apasa tasta **delete** sau **backspace**, elementul sub care se afla cursorul trebuie sa fie sters din lista. Daca lista nu mai contine niciun element comportamentul va fi acelasi cu cel cand utilizatorul apasa tasta **ESC**.
- element neselectat: text normal.
- element selectat: text video invers.
- pozitie cursor: text subliniat.
- element selectat + pozitie cursor: text video invers subliniat.
- Oricare ar fi mijlocul prin intermediul caruia programul termina, configuratia implicita a terminalului **TREBUIE** restaurata. Da, chiar dupa primirea unui semnal

(afara de semanle pe care nu le putem intercepta, dar asta inseamna ca programul vostru nu functioneaza in orice situatie).

- Trebuie sa fie posibil sa se poata intrerupe functionarea programului cu ajutorul combinatiei de taste **ctrl+z** si sa fie restaurat cu **fg** fara ca asta sa influenteze comportamentul sau.
- Daca programul e lansat fara variabile de mediu, trebuie sa adoptati un comportament rezonabil.

## II.2 Bonus

Daca ati reusit sa realizati partea obligatorie in mod perfect, aceasta sectiune propune cateva piste pentru a merge mai departe. Bonusurile vor fi luate in considerare daca obtineti cel putin 18/20 la partea obligatorie.

- Coloanele defileaza de la stanga la dreapta in functie de pozitia cursorului cand fereastra e prea mica.
- Dupa terminarea programului, se sterge ceea ce e de sters, iar prompt-ul si cursorul vor fi afisate pe linia urmatoare de apelul programului. Lansati comanda **tig** pentru a vedea ceea ce vrea sa spun. GAnditi-vă la semnale!
- O interfata frumoasa (corectorul este cel care va judeca, nu voi).
- Daca elementele sunt fisiere, colorati conform extensiilor (un pic ca **ls -G** sub OSX).
- Pozitionarea cursorului in timp ce se tasteaza o secventa de caractere ce corespund unui element al listei (cautare dinamica).

## II.3 Livrare

- Trebuie sa livrati, in radacina directorului vostru de lucru, un fisier **auteur** continand login-l vostru urmat de '\n':
 

```
$>cat -e auteur
xlogin$
```
- Trebuie sa livrati un fisier **Makefile** ce va compila sursele pentru a obtine un executabil denumit **ft\_select**.
- Fisierul **Makefile** trebuie, cel putin, sa propuna regulile **\$(NAME)**, **all**, **clean**, **fclean** si **re** in ordinea in care vi se pare cel mai adevarat.
- Fisierul **Makefile** trebuie sa compileze munca voastră folosind flag-urile de compilare **-Wall**, **-Wextra** si **-Werror**.

- Executabilul trebuie link-editat cu biblioteca `termcap`.
- Doar fisierele ce se gasesc in directorul vostru de lucru vor fi evaluate la sustinere.

## III.4 Consideratii tehnice

- este interzis sa se foloseasca variabilele globale, cu exceptia celor care au fost deja definite pentru voi vous. Se admite o exceptie de la aceasta regula pentru tratarea semnalelor, chiar daca e posibil sa se faca in alt mod.
- O functie locala la un fisier C (care, deci, nu are un prototip in fisierul .h) trebuie definita **static** pentru a respecta prevederile standardului de cod (Norme).
- Trebuie sa fiti atenti la tipuri si sa utilizati in mod judicios operatorul cast cand este necesar, in particular atunci cand un tip **void \*** este implicat. In mod absolut evitati operatorii cast impliciti, indiferent de tipul respectiv.

## III.5 Functii autorizate

- isatty
- ttynname
- ttyslot
- ioctl
- getenv
- tcsetattr
- tcgetattr
- tgetent
- tgetflag
- tgetnum
- tgetstr
- tgoto
- tputs
- open
- close
- write
- malloc
- free
- read
- exit
- signal
- sigaction

# Capitolul III

## Instructiuni

- Puteti folosi functii si fisiere pe care doriti.
- Proiectul vostru trebuie scris conform standardului de cod (Norme). Norminette nu va fi utilizata pentru verificarea Normei care se aplica deci, in ansamblul sau si et va fi verificata de catre o persoana in timpul sustinerii. O eroare de Norma va duce la totarea cu 0 in timpul sustinertii.
- In niciun caz programul vostru nu trebuie sa termine intr-un mod nestepitat (Eroare de segmentare, eroare de magistrala, double free etc.) sau sa intre intr-o bucla infinita. Proiectul va fi considerat ca nefunctional si ca atare va primi nota 0 la sustinere.
- Un proiect care afiseaza de caractere straine si/sau inadecvate este considerat ca nefunctional si va primi nota 0 la sustinere.
- In cazul in care programul vostru este oprit in timpul executiei, terminalul trebuie sa functioneze in mod normal in continuare...
- Toata memoria alocata pe heap trebuie sa fie eliberata in mod corespunzator.
- Trebuie sa livrati, in radacina directorului vostru de lucru/livrare, un fisier **auteur** continand login-ul vostru urmat de '\n':

```
$>cat -e auteur  
xlogin$
```

- Nu trebuie niciodata sa livrati un cod pe care nu l-ati scris voi insiva. In caz ca exista dubii in acest caz, veti fi invitat la o sesiune de recodare pentru a verifica bunele voastre intentii.
- Daca sunteți istet si folositi biblioteca **libft** pentru **ft\_select**, va trebui sa copiati sursele si **Makefile**-ul asociat intr-un director denumit **libft** care trebuie sa se afle in radacina directorului vostru de lucru/livrare.
- Biblioteca **termcap** este obligatorie pentru acest proiect.

- Inainte sa puneti voi intrebarea, aflati ca biblioteca **ncurse** este interzisa a fi folosita la acest proiect. In caz contrar nu ar mai fi nicio provocare...
- Acest proiect necesita o buna refletie. este absolut vital ca sa discutati intre voi si sa partajati informatiile pe care le detineti!

# **Capitolul IV**

## **Notare**

Notarea `ft_select` se face in doi timpi:

- In primul rand partea obligatorie va fi testata. Ea va fi notata cu 20 puncte.
- Apoi, bonusurile vor fi evaluate. Ele vor fi notate cu 22 puncte.
- Bonusurile nu vor fi evaluate decat daca partea obligatorie a fost punctata cu cel putin 18 din 20.
- De asemenea, optimizarea calitatii anumitor elemente de cod vor evaluate, putand astfel sa obtineti puncte suplimentare.

Succes tuturor!