A completely UNIX project

ft_nm, ft_otool

42 staff staff@42.fr

Summary: This project is about recoding the command nm and the command otool

Contents

Chapter I

Foreword

For any finite group G, the order (number of elements) of every subgroup H of G divides the order of G.

Lemma. *Let H be a subgroup of G. Let $r, s \in G$. Then $Hr = Hs$ if and only if $rs^{-1} \in H$, otherwise Hr, Hs have no element in common.*
Similarly, $rH = sH$ if and only if $s^{-1}r \in H$, otherwise rH, sH have no element in common.

Proof. *If $rs^{-1} = h \in H$, then $H = Hh = (Hr)s^{-1}$. Multiplying both sides on the right by s gives $Hr = Hs$.*
Conversely, if $Hr = Hs$, then since $r \in Hr$ (because $1 \in H$) we have $r = h's$ for some $h' \in H$. Multiplying on the right by s^{-1} shows that $rs^{-1} \in H$.

Now suppose Hr, Hs have some element in common, that is $h_1 r = h_2 s$ for some $h_1, h_2 \in H$. This implies $rs^{-1} = h_1^{-1} h_2 \in H$, thus $Hr = Hs$ by above.

Lagrange's Theorem. *If H is a subgroup of G then $|G| = n|H|$ for positive integer n. This is called the index of H in G. Furthermore, there exist $g_1, ..., g_n$ such that $G = Hr_1 \cup ... \cup Hr_n$ and similarly with the left-hand cosets relative to H*

Proof. *Take any $r_1 \in G$. Note $|Hr_1| = |H|$. If $Hr_1 \neq G$ then take any $r_2 \in G \setminus Hr_1$. By the lemma, Hr_1, Hr_2 are disjoint so we have $|Hr_1 \cup Hr_2| = 2|H|$. By continuing in this fashion, after n steps for some positive integer n, we will eventually have accounted for all of the elements of G. We will have $|G| = n|H|$ and $G = Hr_1 \cup ... \cup Hr_n$.*

Corollary. *Let G be a group and $g \in G$. Then the order of g divides $|G|$*

Corollary. *Let G be a group of prime order. Then G has no subgroups and hence is cyclic.*

Chapter II

General Instructions

- This project will be corrected by humans only. You're allowed to organise and name your files as you see fit, but you must follow the following rules.

- The executable must be named `ft_nm` and `ft_otool`

- You must use C and submit a Makefile.

- Your `Makefile` must compile the project and must contain the usual rules. It must recompile and re-link the program only if necessary.

- If you are clever, you will use your library for your `mod1`. Submit also your folder `libft` including its own `Makefile` at the root of your repository. Your `Makefile` will have to compile the library, and then compile your project.

- Your project must be written in accordance with the Norm. Only norminette is authoritative.

- You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).

- You'll have to submit a file called `author` containing your usernames followed by a '\n' at the root of your repository.

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

- Within the mandatory part, you are allowed to use the following functions:

  - open(2)
  - close(2)
  - mmap(2)
  - munmap(2)
  - write(2)

    - fstat(2)
    - malloc(3)
    - free(3)

- You can ask your questions on the forum, on slack...

Chapter III

Mandatory part

You have to recode the `nm` (with no options) and the `otool` command (exactly the same as `otool -t`)

```
$ man nm

$ man otool
```

Chapter IV

Bonus part

⚠ We will look at your bonuses if and only if your mandatory part is EXCELLENT. This means that your must complete the mandatory part, beginning to end, and your error management must be flawless, even in cases of twisted or bad usage. If that's not the case, your bonuses will be totally IGNORED.

As bonus, you can do nm and otool's options.