MY_SUDO

MAKE ME A SANDWICH

MY_SUDO

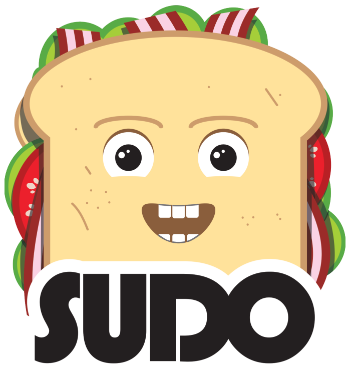
binary name: my_sudo
language: C
compilation: via Makefile, including re, clean and fclean rules
Authorized functions: All the libC excepted: getpw*, *getsp*, getgr*, *fork*, *clone*, openat*

- ✓ The totality of your source files, except all useless files (binary, temp files, objfiles,...), must be included in your delivery.
- ✓ All the bonus files (including a potential specific Makefile) should be in a directory named bonus.
- ✓ Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

You are asked to re-implement the `sudo` command.

Your program must be able to be invoqued as such:

```
~/B-PSU-100> ./my_sudo -h
usage: ./my_sudo -h
usage: ./my_sudo [-ugEs] [command [args ...]]
```



Sudoers

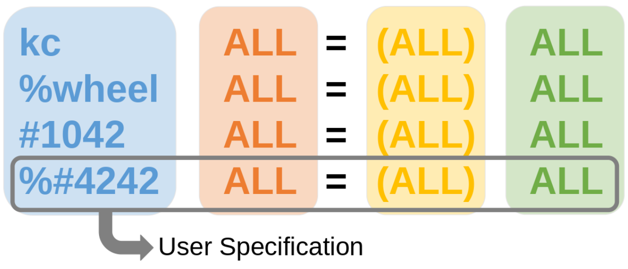
Your program must read its security policy from `/etc/sudoers`.

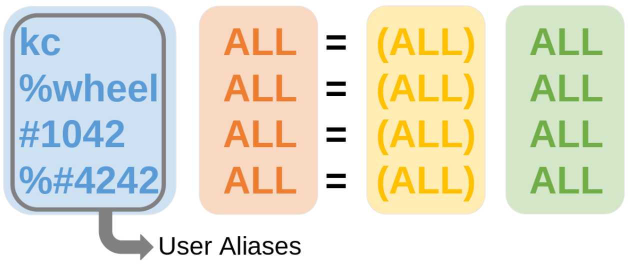
```
man sudoers
```

Example:

Considering the following sudoers file:

```
Terminal
~/B-PSU-100> cat /etc/sudoers
...
kc ALL=(ALL) ALL
%wheel ALL=(ALL) ALL
#1042 ALL=(ALL) ALL
%#4242 ALL=(ALL) ALL
...
```





For this project you must handle only User Aliases in the sudoers file.
(ie: Always consider that everything after the User Alias is set to “ALL=(ALL) ALL” for the autograder)
However, being able to handle the entire User Specification would be a nice bonus.

Terminal

```
~/B-PSU-100> whoami
kc
~/B-PSU-100> ./my_sudo /usr/bin/whoami
[my_sudo] password for kc:
root
```

Terminal

```
~/B-PSU-100> whoami
kc
~/B-PSU-100> ./my_sudo -u toto whoami
[my_sudo] password for kc:
toto
~/B-PSU-100> ls -l top_secret
total 4
-rw-r---. 1 toto admins 3 Jun 27 12:09 top_secret
~/B-PSU-100> cat top_secret
cat: top_secret: Permission denied
~/B-PSU-100> ./my_sudo -u toto cat top_secret
42
```

Don’t hesitate to create some dummy users and groups on your computer in order to toroughly test your program.

You may also want to modify your `/etc/sudoers` file, but be careful to not break your GNU/Linux installation. Try to find a safe way to edit this file.

More examples

Using the same /etc/sudoers file.

```
Terminal
~/B-PSU-100> whoami
kc
~/B-PSU-100> groups toto
toto : admins docker
~/B-PSU-100> ./my_sudo -u toto yes
[my_sudo] password for kc:
~/B-PSU-100> ps -C yes -o ruser,rgroup,euser,egroup
COMMAND RUSER RGROUP EUSER EGROUP
yes toto admins toto admins
~/B-PSU-100> ./my_sudo -u toto -g docker yes
[my_sudo] password for kc:
~/B-PSU-100> ps -C yes -o ruser,rgroup,euser,egroup
COMMAND RUSER RGROUP EUSER EGROUP
yes toto docker toto docker
```

```
Terminal
~/B-PSU-100> whoami
tata
~/B-PSU-100> groups tata
tata : tata audio cdrom video wheel
~/B-PSU-100> ./my_sudo whoami
[my_sudo] password for tata:
root
```

```
Terminal
~/B-PSU-100> whoami
tutu
~/B-PSU-100> groups tutu
tutu : tutu audio cdrom video
~/B-PSU-100> ./my_sudo rm -rf /*
[my_sudo] password for tutu:
tutu is not in the my_sudoers file.
```

Terminal

```
~/B-PSU-100> id
uid=1009(titi) gid=1009(titi) groups=1009(titi),1042(users),4242(birds)
~/B-PSU-100> ./my_sudo whoami
[my_sudo] password for titi:
root
```

Terminal

```
~/B-PSU-100> whoami
kc
~/B-PSU-100> env | grep -E "^SHELL="
SHELL=/bin/bash
~/B-PSU-100> ./my_sudo -s
[my_sudo] password for kc:
~/B-PSU-100> ps -p $$ -o cmd,user,group
CMD USER GROUP
/bin/bash root root
~/B-PSU-100> exit
~/B-PSU-100> whoami
kc
~/B-PSU-100> export SHELL=/bin/sh
~/B-PSU-100> ./my_sudo -s
[my_sudo] password for kc:
~/B-PSU-100> ps -p $$ -o cmd,user,group
CMD USER GROUP
/bin/sh root root
~/B-PSU-100> exit
~/B-PSU-100> whoami
kc
~/B-PSU-100> unset SHELL
~/B-PSU-100> grep -E "^kc|^toto" /etc/passwd
kc:x:1000:1000:KC:/home/kc:/bin/zsh
toto:x:1001:4200::/home/toto:/bin/bash
~/B-PSU-100> ./my_sudo -s -u toto
[my_sudo] password for kc:
~/B-PSU-100> ps -p $$ -o cmd,user,group
CMD USER GROUP
/bin/zsh toto admins
```

User authentication

You must be able to authenticate users invoquing the my_sudo command with their password.

`man shadow`, `man crypt`

In case of an incorrect password entry you should allow the user of your program to make two additional attempts.
Once the user password is entered correctly you should print the amount of unsucessful attempts.

Unlike the original sudo command, the password will be read directly from stdin and not via a pty.

You are not allowed to use the PAM (Pluggable Authentication Module) library to handle user authentication.

Security

With great power, there must also come great responsibility.
— Ben Parker —

Since sudo is such a powerful tool, make sure that your implementation is safe to use.

Your program should not disclose information on the user credentials upon password entering.

And remember that user input should not be trusted.

Bonus

Here are some bonus ideas:
- Complete User Specification compatibility
- `-l`, `--list` flag
- Logging

v1

{EPITECH}