

5.1)

Si on essaie de Unlock un fichier qui n'a pas de verrou, le fichier reste inchangé, il ne se passe rien sauf que `l_type` deviendra `== F_UNLCK`, c'est en fait ce qu'on fait avec l'appel `f_getlk`, quand le lock est possible, on transforme le `l_type` en `f_unlock`, et il ne se passe rien sauf que `l_type` sera modifié et sera `== F_UNLCK`.

5.2)

a) Le résultat dépend des types de verrous qu'on veut placer, si les deux sont des verrous type exclusif, alors on ne pourra pas placer le 2eme verrou, dans le cas où les deux sont des verrous type partage, alors on pourra effectivement en mettre tant qu'on veut (c'est le but même). Cependant, si le premier verrou est de type partagés (read), alors on ne pourra pas placer d'autres verrous de type exclusifs (write).

b) En mettant deux verrous depuis le même processus/terminal, on remplace les éléments dans la structure flock de notre 1^{er} lock par le nouveau input de l'utilisateur du coup on abandonne le premier en le remplaçant par le deuxième, ce qui implique que depuis le même terminal on arrive pas à mettre 2 locks, peu importe leurs type. Dans le cas où on a 2 processus différents, il se passe ce qui a été décrit dans le point a). En plus de cela, le 2^{ème} processus retournera -1 et les erreur EACCES ou EAGAIN pour `F_SETLCK`, l'erreur sera EINTR pour `F_SETLCKW` si on reçoit un signal pendant que celui-ci attend (lors d'un conflit) pour lock une partie du fichier.