# Упражнение 6

1. Да се пуснат един след друг два системни примитива fork(), като за всеки генериран процес се отпечатва pid-а на процеса/getpid()/ и pid-a на родителския ппроцес/getppid()/.
2. Да се пусне един fork() и само в детския процес още един fork(). За всеки генериран процес се отпечатва pid-а на процеса/getpid()/ и pid-a на родителския ппроцес/getppid()/.
3. Да се пуснат три fork() един след друг. Колко процеса се генерират?
4. Да се напише програма на С, която получава като параметри две имена на файлове. Отваря първия файл за писане (ако не съществува, го създава, в противен случай го занулява), създава 2 процеса. Родителят пише hello1, а детето – hello2 във първия файл.
5. В зад 4 пренасочвате изхода на детето във втория файл. В общи действия на двата процеса се записват hello3 на стандартния изход и hello4 във втория файл:
	* с wait в родителя
	* без wait в родителя
6. В зад 5 в края на действията на детето възстановявате стандартния му изход.
7. В зад 6 – да се пробва с exit() – в единия или двата процеса; с или без wait()
8. Да се напише програма на C, която получава като параметри от команден ред две команди (без параметри). Изпълнява първата. Ако тя е завършила успешно изпълнява втората. Ако не, завършва с код -1.
9. Да се напише програма на C, която получава като командни параметри две команди (без параметри). Изпълнява ги едновременно и извежда на стандартния изход номера на процеса на първата завършила успешно. Ако нито една не завърши успешно извежда -1.
10. Да се напише програма на C, която получава като параметри от команден ред две команди (без параметри) и име на файл в текущата директория. Ако файлът не съществува, го създава. Програмата изпълнява командите последователно, по реда на подаването им. Ако първата команда завърши успешно, програмата добавя нейното име към съдържанието на файла, подаден като команден параметър. Ако командата завърши неуспешно, програмата уведомява потребителя чрез подходящо съобщение.
11. Трасировка – upr6\_tras1; upr6\_tras2; upr6\_tras3