

# Упражнение 4

## Системни функции за процеси

```
#include <unistd.h>
int execl(const char *path, const char *arg, ...) - сменя образа на процеса
int execvp(const char *file, const char *arg, ...) - сменя образа на процеса
int execv(const char *path, char *const argv[]) - сменя образа на процеса
int execvp(const char *file, char *const argv[]) - сменя образа на процеса

#include <sys/types.h>
#include <unistd.h>
pid_t fork(void) - поражда нов процес наследник

#include <sys/types.h>
#include <sys/wait.h>
pid_t wait(int *status) - изчаква завършването на процес наследник
pid_t waitpid(pid_t pid, int *status, int options) - изчаква завършването на процес наследник

WNOHANG - не чака завършване на наследник, ако няма - връща 0
WIFEXITED(status) - макрос за нормално завършил наследник
WEXITSTATUS(status) - макрос за статус на нормално завършил наследник
WIFSIGNALED(status) - макрос за наследник прекъснат от сигнал
WTERMSIG(status) - макрос за сигнал прекъснал наследника

#include <stdlib.h>
void exit(int status) - завършва процес

#include <sys/types.h>
#include <unistd.h>
pid_t getpid(void) - получава номера на текущия процес
pid_t getppid(void) - получава номера на родителския процес
```