



OS Lab 10

▼ Status	approved
☑ checkbox	☑
▼ class	OS
📅 due date	@Mar 24, 2021

Task



10. Код завершения команды

Напишите программу, которая запускает команду, заданную в качестве первого аргумента, в виде порожденного процесса. Все остальные аргументы программы передаются этой команде. Затем программа должна дождаться завершения порожденного процесса и распечатать его код завершения.

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <wait.h>

int main (int argc, char **argv){
    if (argc < 2){
        printf("not enough arguments\n");
        return 0;
    }

    int status;
    pid_t pid, ret;

    if ((pid = fork()) == -1){
        perror("fork failed: ");
        return 1;
    }

    if (pid == 0){
        if (execvp(argv[1], &argv[1]) == -1){
            perror("failed to execute: ");
            return 1;
        }
    }

    ret = wait(&status);

    if (ret == -1){
        perror("wait(2) error: ");
        return 1;
    }

    printf("exit status: %d\n", WEXITSTATUS(status));
    return 0;
}
```

Когда процесс ожидает получения слова состояния своих подпроцессов с использованием `wait(2)` или `waitpid(3C)`, то это слово может быть проанализировано при помощи макросов, определенных в `<sys/wait.h>`. Эти макросы обсуждаются на странице руководства `wstat(5)`.

`WIFEXITED(stat)` Ненулевое значение, если это слово состояния получено от подпроцесса, завершившегося по `exit(2)`.

`WEXITSTATUS(stat)` Если значение `WIFEXITED(stat)` ненулевое, этот макрос возвращает код завершения, переданный подпроцессом вызову `exit(2)`, или возвращенный его функцией `main()`, иначе код возврата не определен.

`WIFSIGNALED(stat)` Возвращает ненулевое значение, если это слово состояния получено от подпроцесса, который был принудительно завершён сигналом.

`WTERMSIG(stat)` Если значение `WIFSIGNALED(stat)` ненулевое, этот макрос возвращает номер сигнала, который вызвал завершение подпроцесса, иначе код возврата не определен.

`WIFSTOPPED(stat)` Возвращает ненулевое значение, если слово состояния получено от приостановленного подпроцесса (`wait(2)` не реагирует на приостановленные подпроцессы, такое слово состояния может быть получено только вызовом `waitpid(2)`).

`WSTOPSIG(stat)` Если значение `WIFSTOPPED(stat)` ненулевое, этот макрос возвращает номер сигнала, который вызвал приостановку подпроцесса, иначе код возврата не определен.

`WIFCONTINUED(stat)` Возвращает ненулевое значение, если слово состояния

получено от процесса, продолжившего исполнение (`wait(2)` не реагирует на приостановленные подпроцессы, такое слово состояния может быть получено только вызовом `waitpid(2)`).

`WCOREDUMP(stat)` Если значение `WIFSIGNALED(stat)` ненулевое, этот макрос возвратит ненулевое значение, если был создан посмертный дамп памяти (`coredump`-файл) завершившегося подпроцесса. Факт создания дампа памяти определяется по номеру сигнала; завершение по некоторым сигналам, таким, как `SIGSEGV` и `SIGFPE`, всегда приводит к созданию дампа памяти, завершение по остальным сигналам никогда не создает такой дамп.

Notes

Reading list

