**ЗАДАНИЕ**

**LAB 5.2**

**Процесс в состоянии зомби** — это процесс, который завершился (выполнил свой код), но его структура данных (информация о завершении) все еще остаётся в ядре операционной системы, потому что родительский процесс не вызвал wait() или waitpid() для того, чтобы получить код завершения дочернего процесса.

*a. Модифицируйте предыдущую программу так чтобы дочерний процесс становился зомби.*

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

int main() {

pid\_t pid = fork();

if (pid == 0) {

printf("Дочерний PID: %d\n", getpid());

exit(0);

} else {

printf("Родительский PID: %d\n", getpid());

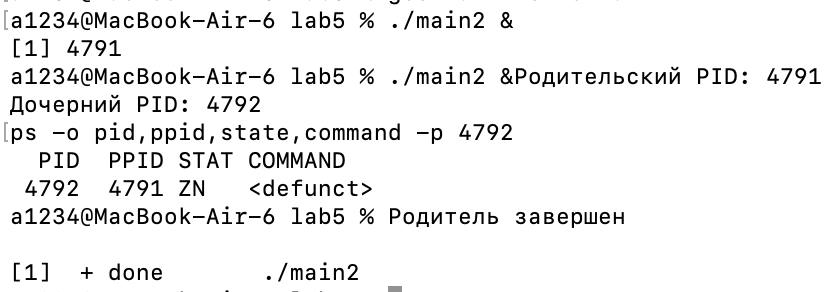
sleep(30);

printf("Родитель завершен\n");

}

return 0;

}



*b. Объясните какую проблему решает данное состояние.*

1. Решение проблемы отслеживания завершения процесса:

Когда процесс завершается, операционная система должна сохранить информацию о его завершении, чтобы родительский процесс мог узнать, как он завершился (успешно или с ошибкой), а также получить код возврата. Состояние зомби позволяет сохранить эту информацию. Операционная система должна оставить информацию о завершении процесса в таблице процессов, чтобы родительский процесс мог потом получить эту информацию. Родитель должен быть уведомлён, что его потомок завершился, и только после этого операционная система может удалить процесс из таблицы.

Зомби-процесс — это своего рода метка, которая говорит о том, что процесс завершился, но информация о завершении ещё не была обработана родителем. После того как родитель вызовет wait() или waitpid(), процесс будет окончательно удалён.

2. Решение проблемы с многозадачностью и асинхронной обработкой:

Механизм зомби-процессов также решает проблему асинхронной обработки завершения процессов. Родитель может не сразу обрабатывать завершение дочернего процесса, например, если он занят выполнением других задач или ждал другого события. Зомби позволяет этим процессам остаться в системе и быть восстановленными позже, когда родительский процесс вызовет wait().

*c. Может ли родительский процесс оказаться в состоянии зомби? Если да,*

*то что в этом случае произойдет с дочерним? Смоделируйте эту*

*ситуацию.*

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

int main() {

pid\_t pid = fork();

if (pid == 0) {

sleep(20);

printf("Дочерний PID: %d (PPID теперь: %d)\n", getpid(), getppid());

exit(0);

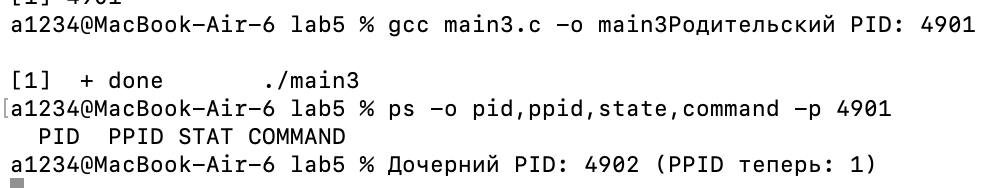
} else {

printf("Родительский PID: %d\n", getpid());

exit(0);

}

}



• Родительский процесс не может быть зомби — зомби это всегда дочерний процесс.

• Если родитель завершается раньше дочернего, тот становится сиротой и наследуется init, который корректно обработает его завершение.