

Basic Theory:

In part 1 when we call fork it creates a new similar to parent process. Fork copies all the variables of the parent process(like open files, signal handlers and signal dispositions, current working directory, user and group IDs.) after doing fork child and parent are 2 different processes(their PID also differ).

But incase of pthread_create it creates new thread in the program with the same PID. This new thread and parent also share resources like global variables and thread will share data, open files, signal handlers and signal dispositions, current working directory, user and group IDs.

Refer below image: It show what pthread share and what it has own it's own.

Per process vs per thread items

Per process items	Per thread items
Address space	Program counter
Global variables	Registers
Open files	Stack
Child processes	State
Pending alarms	
Signals and signal handlers	
Accounting information	

Question Explanation

As in part 1 it's using a fork so the child and parent process are using 2 different variables so it gives accordingly.

```
Part1: Fork one  
Child -90  
Parent 100
```

Whereas in part2 pthread_create's process and parent process use the same global variable so they give result accordingly.

```
Part 2: Pthread one  
Parent 100  
Child 0
```