

1-

الف) زمانی که پردازنده چندین هسته در اختیار دارد می توان با استفاده از این تکنیک کلایی را به شدت افزایش داد.

ب) در وب سرور ها ، زمانی که هر کلاینت در خواست می دهد باید از تکنیک موازی سازی استفاده کنیم و برای هم کلاینت یک thread ایجاد کنیم یا از threadpool استفاده کنیم یا تاخیر پردازش در خواست کاربران را تا جایی که می شود کاهش داد.

2-

الف) در مواقعی که یک پردازنده تک هسته ای داریم و responsive بودن چندان برای کاربر نیاز نیست بهتر است از single thread استفاده شود زیرا در صورت استفاده از چند thread هم همچنان کلرها sequential پیش می روند و parallelism نداریم .

ب) زمانی که نمی خواهیم سر بار تغییر حالات thread ها (شامل سر بار کپی کردن register های اختصاص داده شده و وضعیت thread و ..) (تغییر حالت شامل wait کردن ، انتقال از wait به running و ...) را نمی خواهیم متحمل شویم و همچنین زمانی که سیستم به شدت احتیاج داد که کار thread کامل تمام شود مانند سیستم عامل های هواپیما و ... می توان از همان تکنیک single thread استفاده کرد.

3-

lineC → result = 5

lineP → result = 0

4-

```
//  
// Created by amirphl on 11/2/18.  
//  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <sys/wait.h>  
int main(int argc, char *argv[]) {  
    pid_t pid;  
    int level = 0;  
    printf("I am in level 0 so level = %d\n", level);  
    pid = fork();  
    if (pid > 0) {  
        level++;  
        printf("I am in level 1 so level = %d\n", level);  
    }  
    pid_t pid_1 = fork();  
    if (pid_1 > 0) {  
        level++;  
        if (pid == 0)  
            printf("I am in level 1 so level = %d\n", level);  
        else  
            printf("I am in level 2 so level = %d\n", level);  
    }  
    pid_t pid_2 = fork();  
    if (pid_2 > 0) {  
        level++;  
        if (pid == 0 && pid_1 == 0)  
            printf("I am in level 1 so level = %d\n", level);  
        else if (pid == 0 && pid_1 > 0)  
            printf("I am in level 2 so level = %d\n", level);  
        else if (pid > 0 && pid_1 == 0)  
            printf("I am in level 2 so level = %d\n", level);  
        else if (pid > 0 && pid_1 > 0)  
            printf("I am in level 3 so level = %d\n", level);  
    }  
    if (pid > 0)  
        wait(NULL);  
    return 0;  
}
```

5-

A) Creates 1023 processes.

B) Creates 23 processes.