



دانشکده مهندسی کامپیوتر

درس سیستم‌های عامل

---

پاسخنامه کوییز اول

---

مدرس ..... دکتر وحید ازهری

طراح ..... فرزانه رحمانی

(1) تفاوت بین هم روندی (Concurrency) و موازی سازی (Parallelism) در روند اجرای سیستم‌ها را با رسم شکل توضیح دهید.

پاسخ:

□ **Parallelism** implies a system can perform more than one task simultaneously → multicore  
موازی سازی به این معنی است که یک سیستم می تواند بیش از یک کار را به طور همزمان انجام دهد

□ **Concurrency** supports more than one task making progress

□ Single processor / core, scheduler providing concurrency

هم روندی از بیش از یک کار در حال پیشرفت پشتیبانی می کند

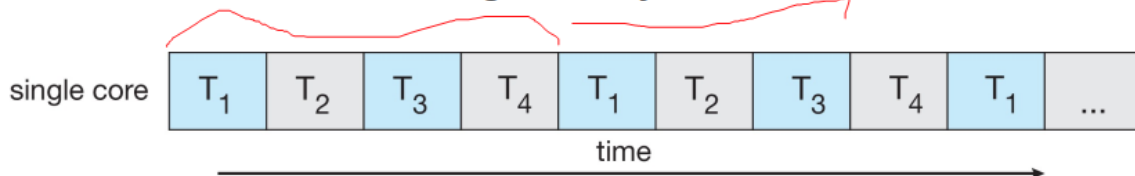


هم روندی

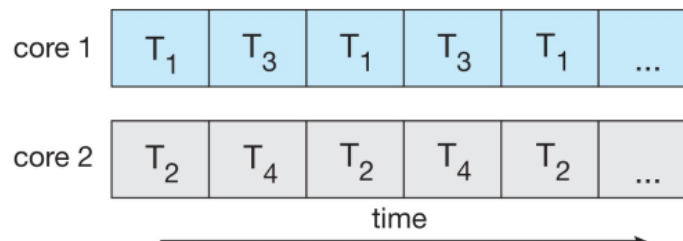
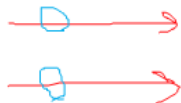
موازی سازی

## Concurrency vs. Parallelism

□ **Concurrent execution on single-core system:** 4 tasks are making progress



□ **Parallelism on a multi-core system:** more than one task simultaneously



۲) کد زیر را با کشیدن درخت فرایندها (processes) تحلیل کنید و بگویید در نهایت چند hello چاپ می شود.

تذکر:

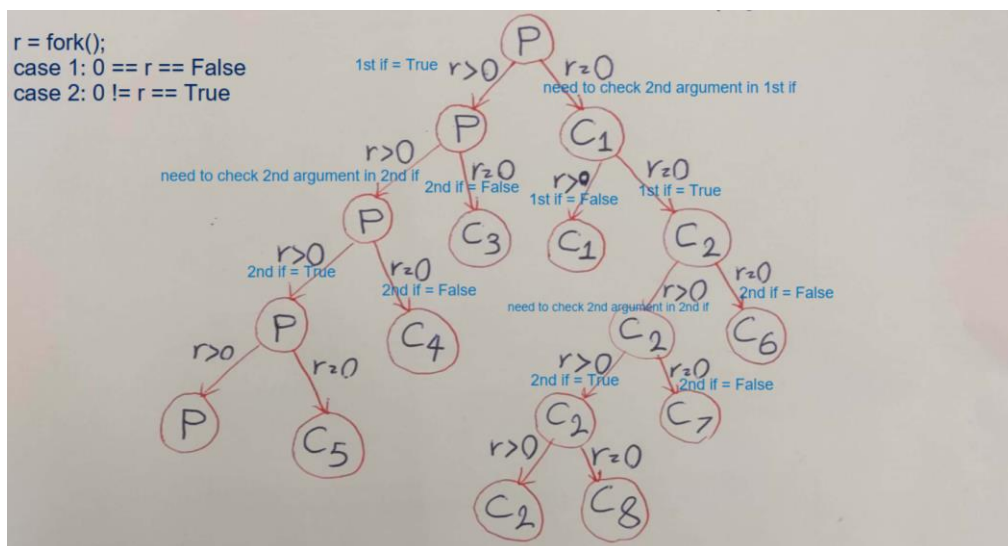
- OR operator (i.e, ||) evaluate second condition when first condition is false.
- AND operator (i.e, &&) if first condition is false then it will not evaluate second condition.
- if(-1) --> true
- if(0) --> false
- if(2) --> true

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

int main(int argc, char *argv[])
{
    if(fork() || (!fork()))
    {
        if(fork() && fork())
        {
            fork();
        }
    }
    printf("hello\n");
    return 0;
}
```

پاسخ:

ابتدا باید توجه کنیم که عملگر  $\&\&$  به گونه ای عمل می کند که اگر عملوند اول غلط باشد، شرط دوم را چک نمی کند و  $\text{fork}$  دوم اجرا نمی شود همچنین عملگر  $\text{or}$  (||) بدین شیوه عمل می کند که اگر شرط اول درست باشد، شرط دوم را محاسبه نمی کند و  $\text{fork}$  دوم اجرا نمی شود. با توجه به این دلایل و درخت فرآیندها در زیر در نهایت عبارت  $\text{"hello"}$  9 بار چاپ می شود.



برگ های درخت فرآیندها در بالا 9 تا هستند و هر کدام یکبار  $\text{"hello"}$  را چاپ می کنند پس در مجموع 9 بار  $\text{"hello"}$  چاپ می شود.

درخت سرم  $\text{fork}$  باعث ایجاد دو  $\text{process}$  می زند که یکی مقدار بزرگتر از 0 دارد پس شرط دوم چک نمی شود ولی دومی که مقدار صفر دارد باعث ایجاد دو  $\text{process}$   $C_2$  و  $C_4$  می شود که چون  $n + 1$  کر قسیم  $C_2$  وارد بلاک شرط می شود. در  $\text{fork}() \& \text{fork}()$  دو پروسس  $P$  و  $C_4$  را  $\text{fork}$  می کند که  $C_3$  چون شرط اولش غلط است  $\text{fork}()$  دوم را اجرا نمی کند و  $P$  دیگری می زند که  $P$  و  $C_5$  را  $\text{fork}$  می کند که  $C_4$  چون غلط است وارد  $\text{if block}$  نمی شود ولی  $P$  می شود و  $C_5$  را  $\text{fork}$  می کند. ~~در  $\text{fork}() \& \text{fork}()$  دو پروسس  $P$  و  $C_4$  را  $\text{fork}$  می کند که  $C_3$  چون شرط اولش غلط است  $\text{fork}()$  دوم را اجرا نمی کند و  $P$  دیگری می زند که  $P$  و  $C_5$  را  $\text{fork}$  می کند که  $C_4$  چون غلط است وارد  $\text{if block}$  نمی شود ولی  $P$  می شود و  $C_5$  را  $\text{fork}$  می کند.~~  $C_2$  نیز  $C_2$  و  $C_6$  را  $\text{fork}$  می کند که  $C_6$  چون غلط است شرط دوم چک نمی شود.  $C_2$  باز  $C_2$  و  $C_7$  را ایجاد می کند که  $C_7$  چون درست باز واحد  $\text{if block}$  می شود و  $C_8$  و  $C_2$  ایجاد می شوند.