

MÔN HỌC: HỆ ĐIỀU HÀNH

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 3

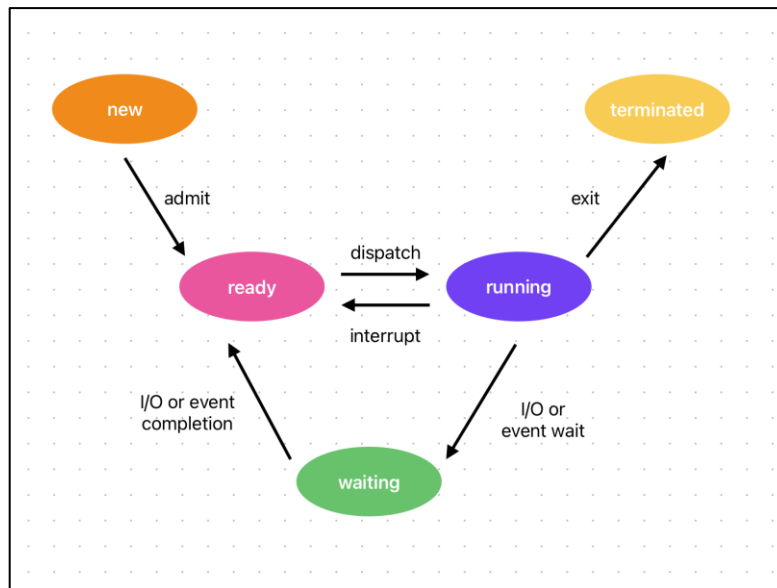
1. Một tiến trình chứa những thành phần gì?

Một tiến trình bao gồm:

- Text section (program code)
- Data section (chứa global variables)
- Program counter, processor registers
- Heap section (chứa bộ nhớ cấp phát động)
- Stack section (chứa dữ liệu tạm thời)
 - + Function parameters
 - + Return address
 - + Local variables

2. Tiến trình có những trạng thái nào? Cách tiến trình chuyển trạng thái?

- Trạng thái tiến trình:
 - + new: tiến trình vừa được tạo
 - + ready: tiến trình đã có đủ tài nguyên, chỉ còn cần CPU
 - + running: các lệnh của tiến trình đang được thực thi
 - + waiting (hay blocked): tiến trình đợi I/O hoàn tất, hoặc đợi tín hiệu
 - + terminated: tiến trình đã kết thúc
- Cách tiến trình chuyển trạng thái:



3. Tại sao phải cộng tác giữa các tiến trình?

Trong tiến trình thực thi, các tiến trình có thể cộng tác (cooperate) để hoàn thành công việc. Các tiến trình cộng tác để:

- + Chia sẻ dữ liệu (information sharing)
- + Tăng tốc tính toán (computational speedup): Nếu hệ thống có nhiều CPU, chia công việc tính toán thành nhiều công việc tính toán nhỏ chạy song song
- + Thực hiện một công việc chung: Xây dựng một phần mềm phức tạp bằng cách chia thành các module/process hợp tác nhau
- + Sự cộng tác giữa các tiến trình yêu cầu hệ điều hành hỗ trợ cơ chế giao tiếp và cơ chế đồng bộ hoạt động của các tiến trình.

4. PCB là gì? Dùng để làm gì?

- Mỗi tiến trình trong hệ thống đều được cấp phát một Process Control Block (PCB).
- PCB là một trong các cấu trúc dữ liệu quan trọng nhất của hệ điều hành.

- PCB gồm:

- + Trạng thái tiến trình: new, ready, running,...
- + Bộ đếm chương trình
- + Các thanh ghi
- + Thông tin lập thời biểu CPU: độ ưu tiên,...
- + Thông tin quản lý bộ nhớ
- + Thông tin: lượng CPU, thời gian sử dụng,
- + Thông tin trạng thái I/O

- Dùng để theo dõi trạng thái tiến trình: PCB làm cho việc quản lý và theo dõi tiến trình trở nên hiệu quả hơn. Mỗi khi một tiến trình được tạo hoặc bị chuyển đổi trạng thái, hệ điều hành sẽ cập nhật thông tin trong PCB tương ứng để đảm bảo rằng quá trình hoạt động của tiến trình được kiểm soát và quản lý một cách chính xác.

5. Tiểu trình là gì?

- Tiểu trình (thread) là một luồng thực thi nhỏ nhất của một chương trình. Nó là một phần nhỏ của tiến trình và chia sẻ cùng không gian bộ nhớ và tài nguyên hệ thống với các tiểu trình khác trong cùng một tiến trình. Một tiến trình có thể chứa một hoặc nhiều tiểu trình.

- Tiểu trình cho phép các tác vụ trong chương trình chạy song song và đồng thời, cải thiện hiệu suất và đáp ứng của chương trình. Mỗi tiểu trình có thể thực hiện một nhiệm vụ cụ thể trong chương trình mà không làm ảnh hưởng đến các tiểu trình khác. Điều này giúp tận dụng tốt các tài nguyên của hệ thống và tối ưu hóa thời gian thực thi.

- Tiến trình là một đơn vị cơ bản sử dụng CPU gồm: Thread ID, PC, Registers, Stack và chia sẻ chung code, data, resources (files)

6. Trình tự thực thi của tiến trình cha và tiến trình con?

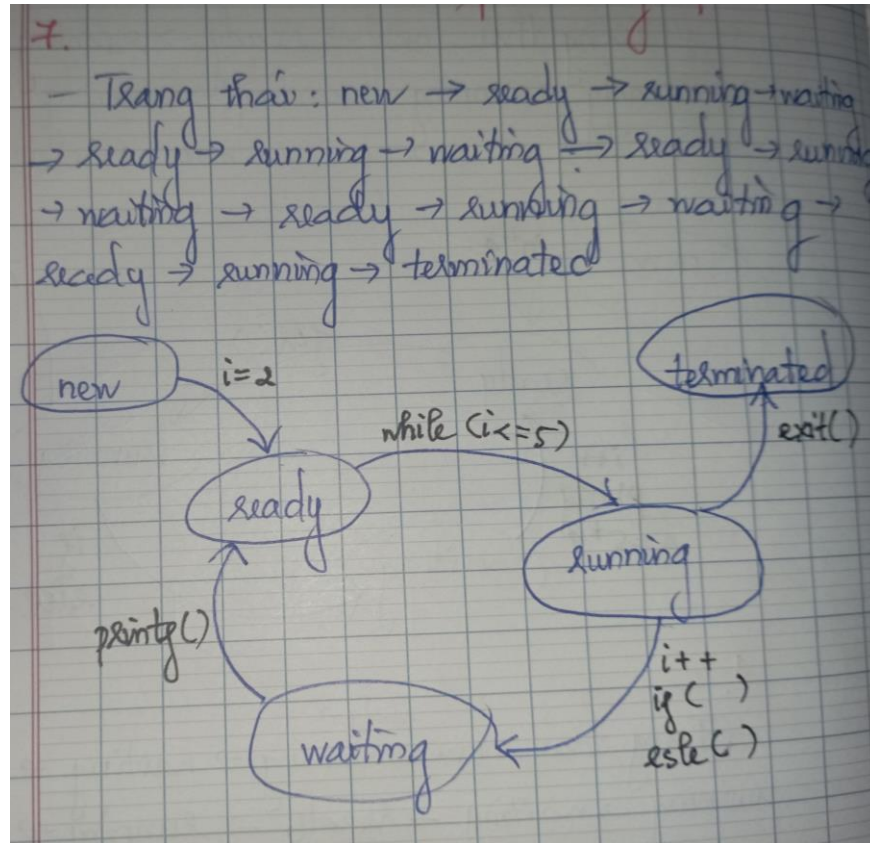
- Tiến trình cha và con thực thi đồng thời (concurrently).
- Tiến trình cha đợi đến khi các tiến trình con kết thúc.

7. (Bài tập mẫu) Cho đoạn chương trình sau:

```
int main (int argc, char** argv)
{
    int i = 2;
    while (i <= 5)
    {
        i++;
        if (i % 2 == 0)
        {
            printf ("Hello");
            printf ("Hi");
        }
        else
        {
            printf ("Bye");
        }
    }
    exit (0);
}
```

Hỏi trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái nào? Vẽ sơ đồ chuyển trạng thái trong quá trình thực thi?

Trả lời:



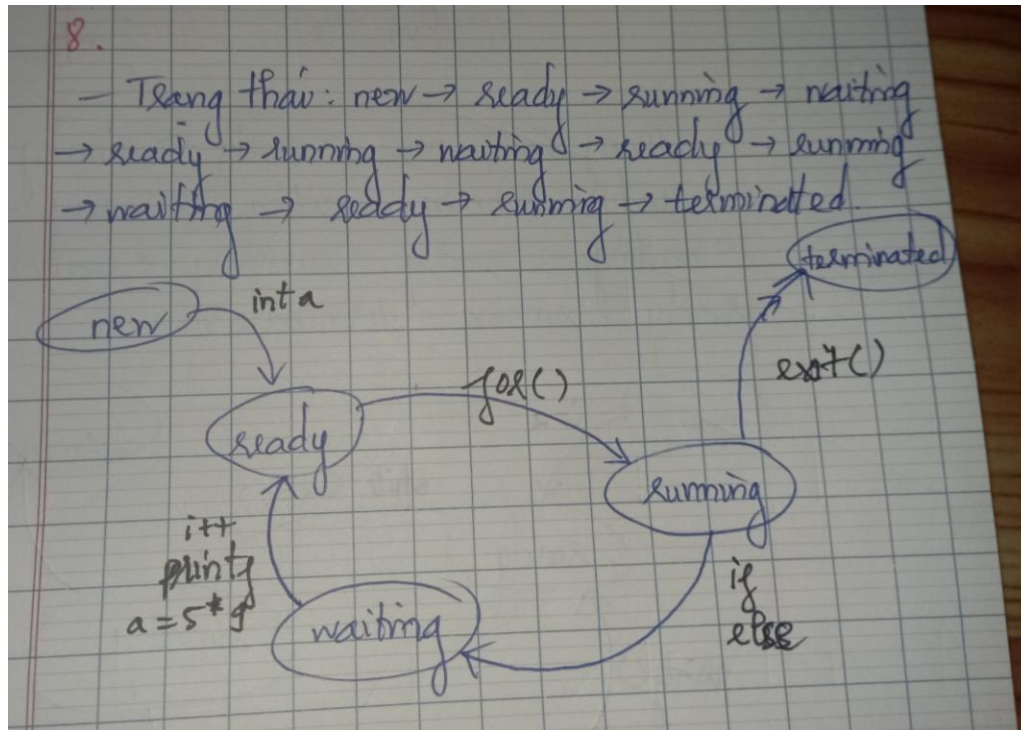
8. Cho đoạn chương trình sau:

```

/* test.c */
int main(int argc, char** argv)
{
    int a;
    for (int i = 1; i < 5; i++)
    {
        if (i % 2 == 0)
            printf("Hello world\n");
        else a = 5*9;
    }
    exit(0);
}

```

Hỏi trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái nào? Vẽ sơ đồ chuyển trạng thái trong quá trình thực thi?



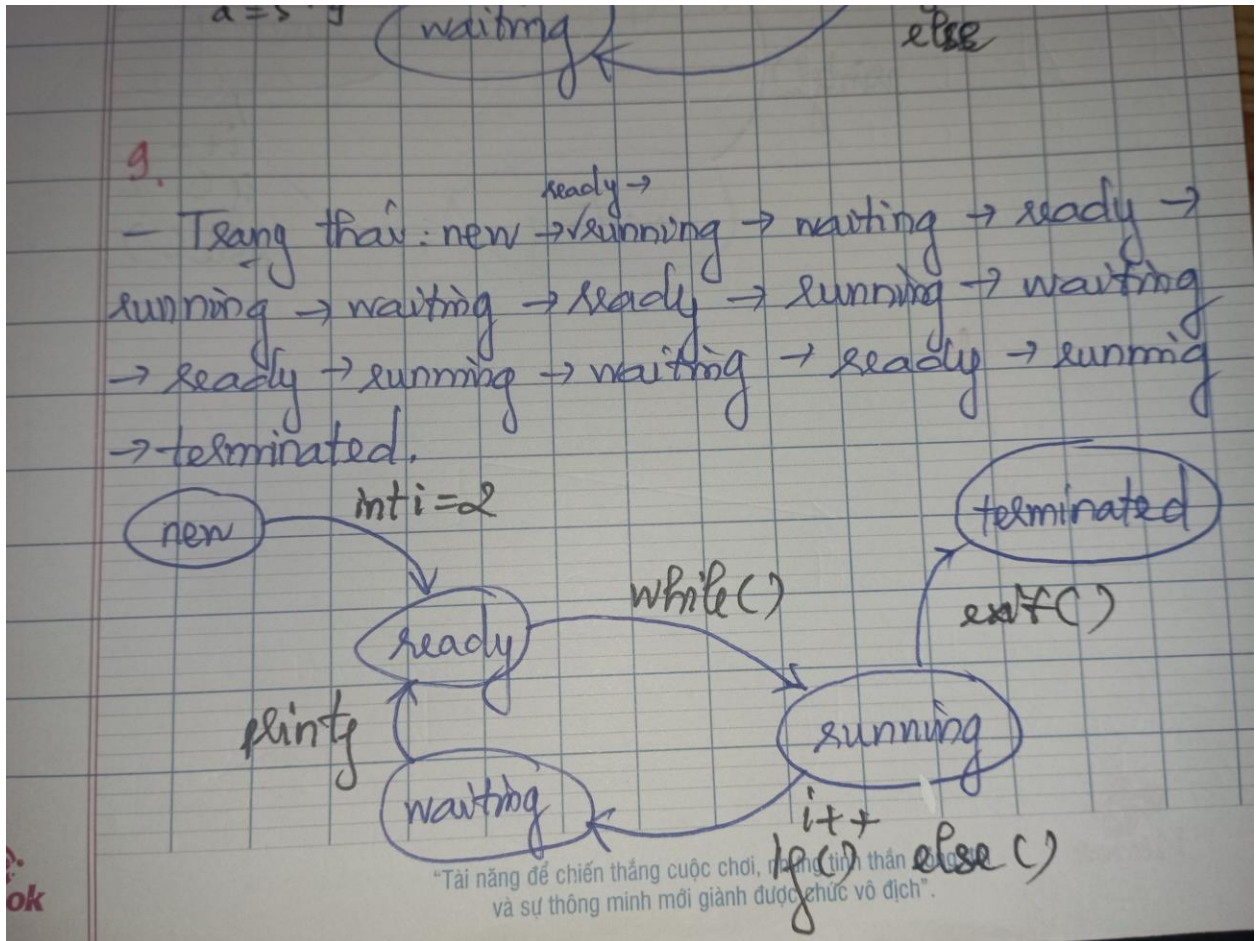
9. Cho đoạn chương trình sau:

```

int main (int argc, char** argv)
{
    int i = 2;
    while (i <= 5)
    {
        i++;
        if (i % 2 == 0)
        {
            printf ("Hello");
            printf ("Hi");
        }
        else
        {
            printf ("Bye");
        }
    }
    exit (0);
}

```

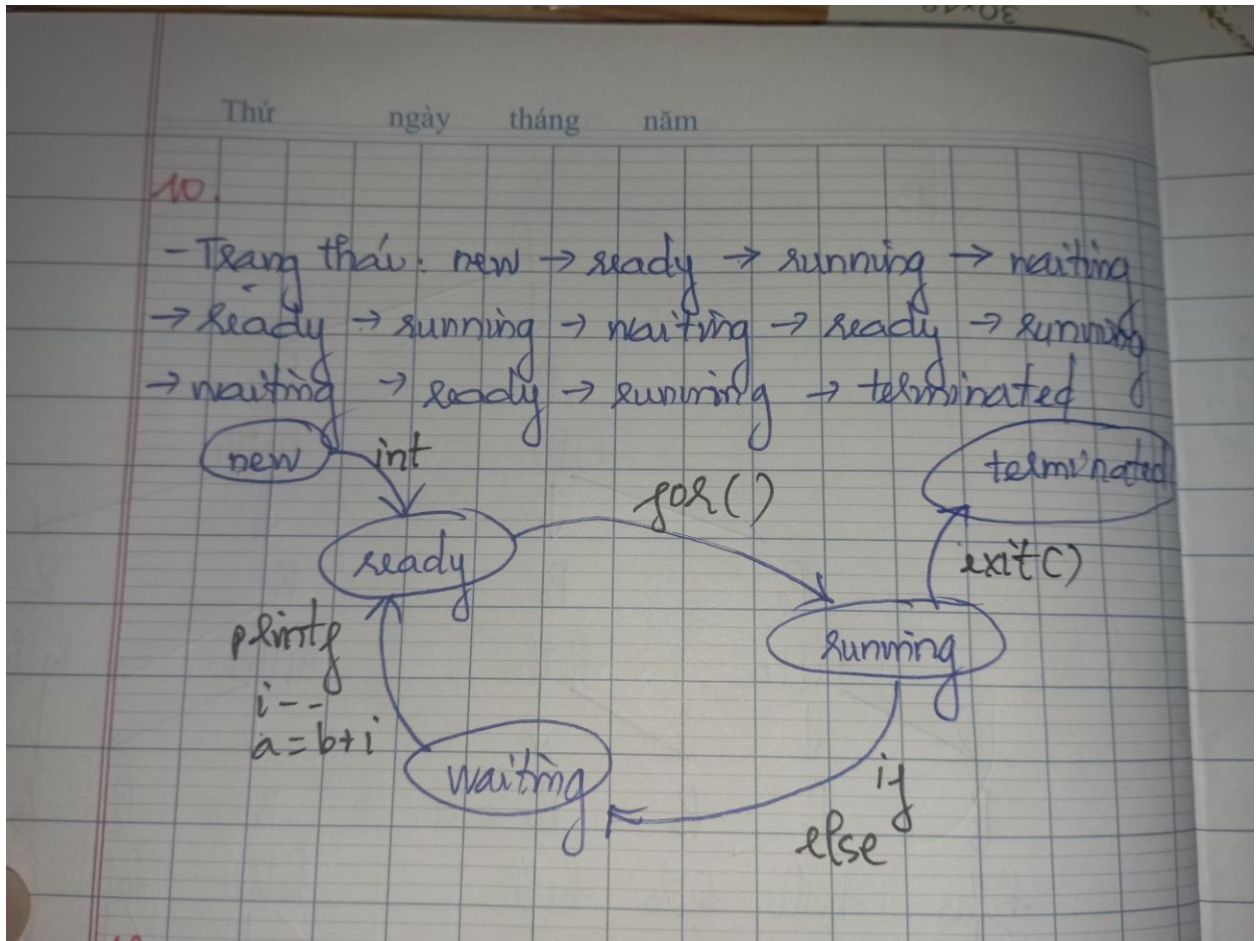
Hỏi trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái nào? Vẽ sơ đồ chuyển trạng thái trong quá trình thực thi?



10. Cho đoạn chương trình sau:

```
int main (int argc, char** argv)
{
    int a, b, i;
    for (i = 16, i >= 6; i --)
    {
        if (i % 3 == 0)
        {
            printf ("Số %d chia hết cho 3", i);
        }
        else
        {
            a = b + i;
        }
    }
    exit (0);
}
```

Hỏi trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái nào? Vẽ sơ đồ chuyển trạng thái trong quá trình thực thi?



11. (Bài tập mẫu) Cho đoạn code sau, hỏi khi chạy, bao nhiêu process được sinh ra và chương trình sẽ in ra những gì? Vẽ cây tiến trình khi thực thi đoạn chương trình sau

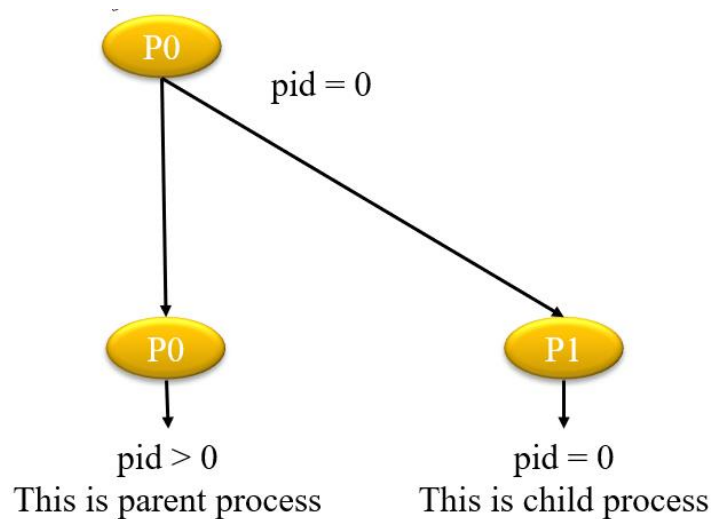
```

#include <stdio.h>
#include <unistd.h>
int main (int argc, char *argv[]){
    int pid;
    /* create a new process */
    pid = fork();
    if (pid > 0){
        printf("This is parent process");
        wait(NULL);
        exit(0);}
    else if (pid == 0) {
        printf("This is child process");
        execlp("/bin/ls", "ls", NULL);
        exit(0);}
    else { // pid < 0
        printf("Fork error\n");
        exit(-1);
    }
}

```

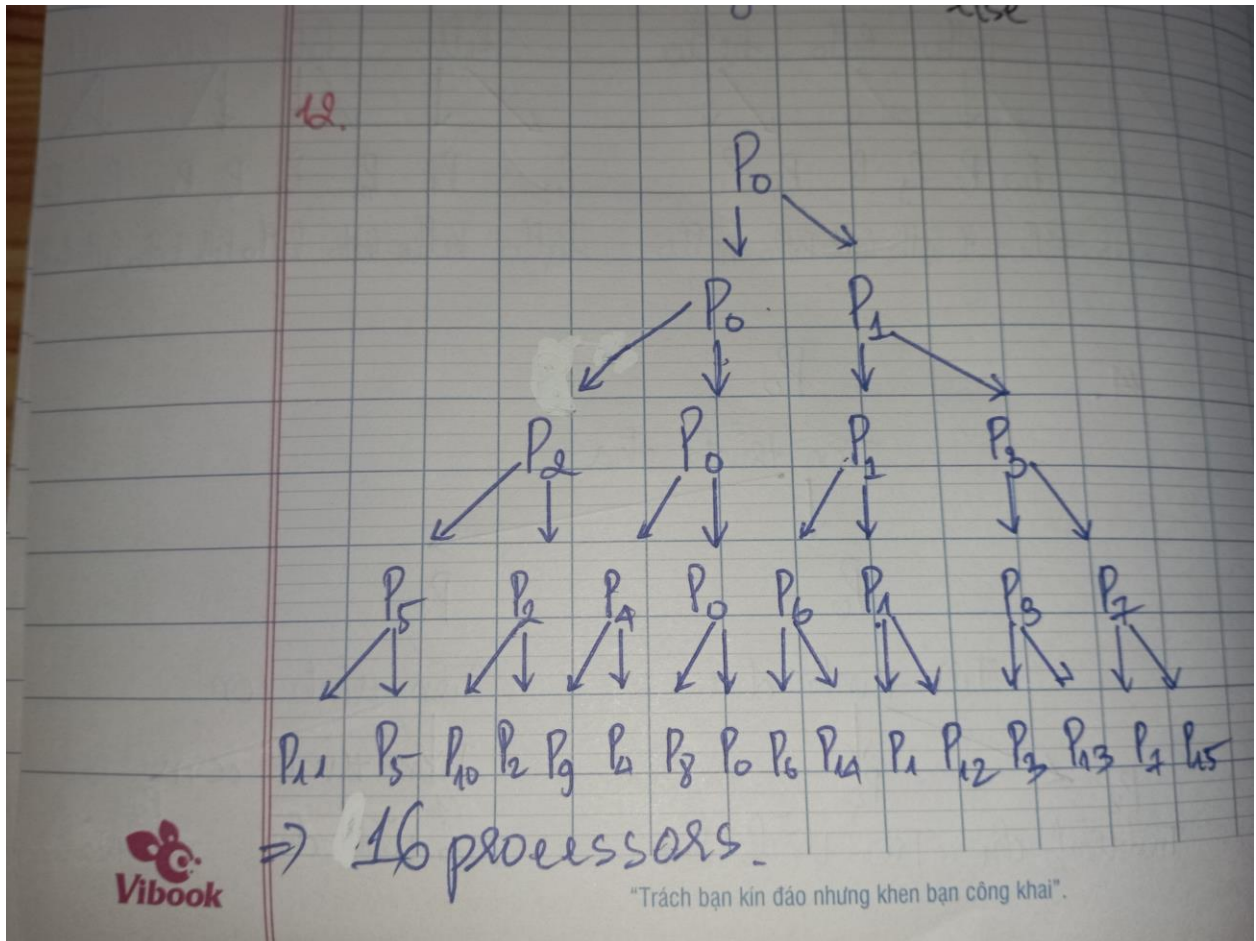
Trả lời:

Khi chạy đoạn chương trình trên, khi chạy hết sẽ có 2 process được sinh ra bao gồm 1 tiến trình cha và 1 tiến trình con. Theo chương trình trên thì tiến trình cha sẽ in ra dòng chữ “This is parent process”; và tiến trình con sẽ in ra dòng chữ “This is child process”. Cây tiến trình khi thực thi đoạn chương trình trên như sau:



12. Cho đoạn code sau, hỏi khi chạy, bao nhiêu process (kể cả cha) được sinh ra? Vẽ cây tiến trình khi thực thi đoạn chương trình sau

```
int main()
{
    fork();
    fork();
    fork();
    fork();
    return 0;
}
```

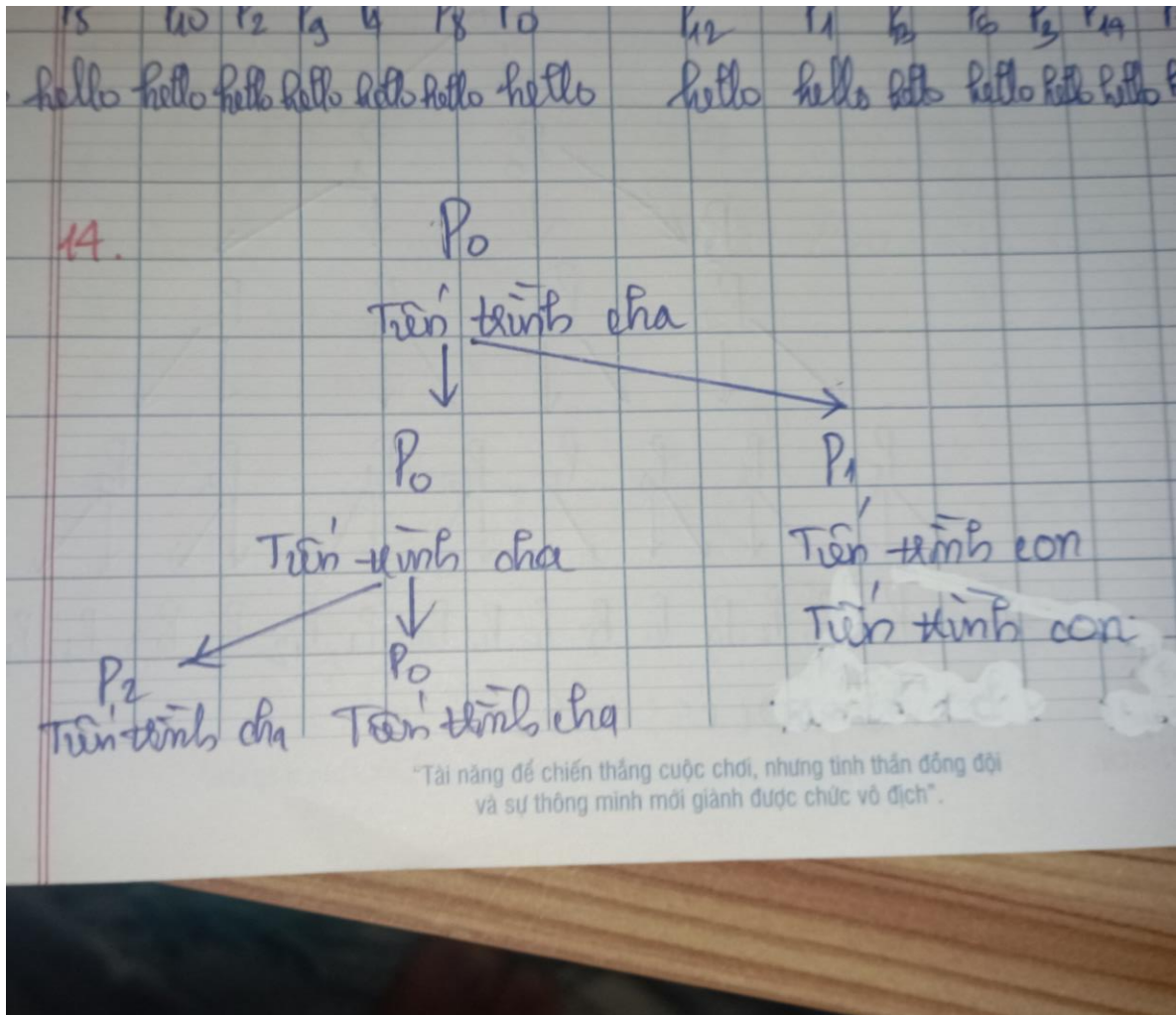



13. Cho đoạn code sau, hỏi khi chạy thì tiến trình được tạo ra từ chương trình trên sẽ in ra màn hình những gì? Vẽ cây tiến trình và những từ được in ra khi thực thi đoạn chương trình sau?

```

#include <stdio.h>
#include <unistd.h>
int main()
{
    int i;
    for (i = 0; i < 4; i++)
    {
        fork();
        printf("hello\n");
    }
    return 0;
}

```

15. Cho đoạn code chương trình sau:

```

if (fork() == 0)
{
    a = a + 5;
    printf("%d,%d\n", a, &a);
}
else
{
    a = a - 5;
    printf("%d, %d\n", a, &a);
}
  
```

Giả sử u, v là các giá trị được in ra bởi process cha, và x, y là các giá trị được in ra bởi process con. Tìm mối quan hệ giữa u, v và x, y?

Thứ ngày tháng năm

15.

P_0



P_0



P_1

— P_0 (cha):

a tăng 5

đ/c đk đv?

— P_0 (con):

a giảm 5

đ/c đk đv?