

1. What are the advantages of Makefile? Give examples?

- Trong một dự án lớn, việc xây dựng lại chương trình có thể mất một khoảng thời gian lớn vì sẽ có nhiều tệp được biên dịch và liên kết. Khi làm việc trên dự án và thực hiện một chút thay đổi, điều đó sẽ gây khó chịu. mỗi lần phải đợi xây dựng lại mọi thứ.

- Makefile giúp giảm thời gian biên dịch khi có sự thay đổi của các file thành phần do không phải biên dịch lại mọi thứ dù là những file không liên quan đến sự thay đổi.

- Làm cho cấu trúc project mạch lạc, dễ đọc và dễ debug.

- Makefile sẽ tự động compile những file có thay đổi và phụ thuộc vào sự thay đổi đó

2. In case of source code files located in different places, how can we write a Makefile?

Ta có thể dùng cách này để khai báo đường dẫn với chương trình

-obj/%.o:%.c

-gcc -c -o \$@ \$(CPPFLAGS) \$(CFLAGS) \$<

Nó bảo với make rằng, e.g. obj/file0.o được tạo từ file0.c theo nguyên tắc:

gcc -c -o obj/file0.o \$(CPPFLAGS) \$(CFLAGS) file0.c

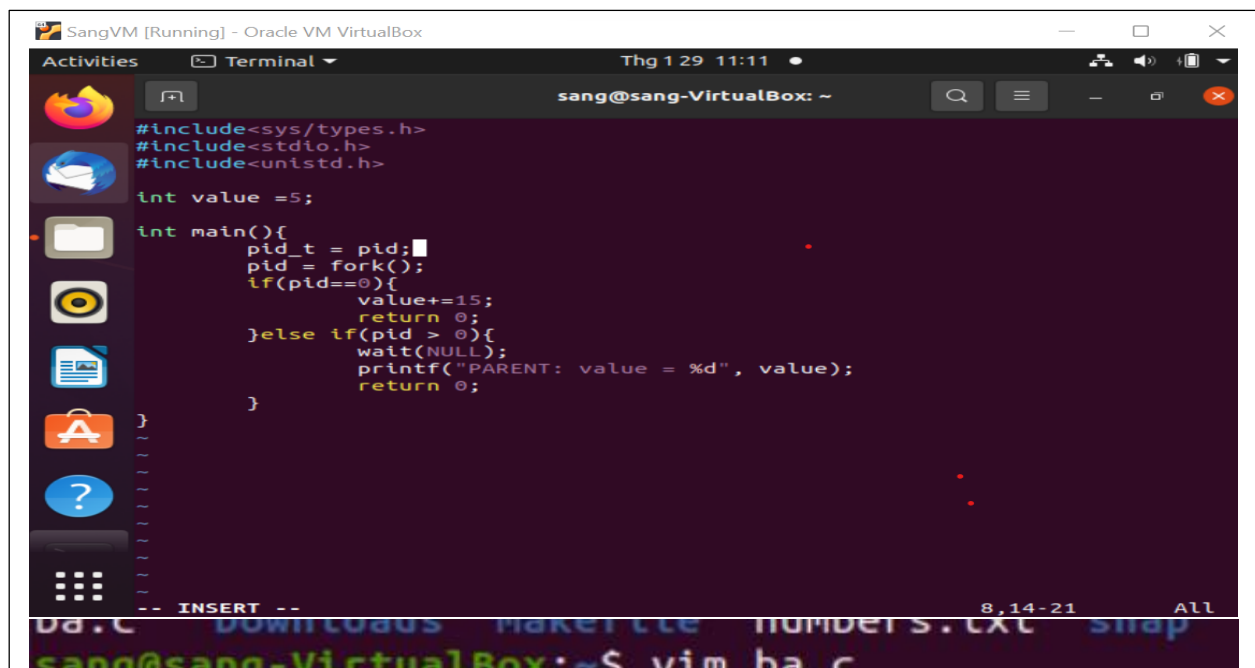
3. What the output will be at LINE A? Explain your answer.

Chương trình sẽ in ra màn hình **"PARENT: value = 5"**

Vì: Chương trình gọi hàm fork() sẽ tạo ra một process con hoạt động độc lập với process cha. Nên mọi sự thay đổi của process con cụ thể là value += 15 sẽ không liên quan đến process cha và kết quả chương trình.

Nên khi process cha được thực thi sẽ in ra giá trị value toàn cục được khởi tạo bằng 5.

Kết quả kiểm thử:



```
#include<sys/types.h>
#include<stdio.h>
#include<unistd.h>

int value =5;

int main(){
    pid_t = pid;
    pid = fork();
    if(pid==0){
        value+=15;
        return 0;
    }else if(pid > 0){
        wait(NULL);
        printf("PARENT: value = %d", value);
        return 0;
    }
}
```