LẬP TRÌNH C TRÊN LINUX

1. **Giới thiệu về lập trình C trên Linux**

   Đa số mọi người bắt đầu lập trình C đều bằng Turbo C, Borland C, Visual Studio C. Chúng là các ứng dụng trên DOS hoặc Windows. Vì thế mà chúng trở thành quen thuộc với những người mới tiếp cận, dẫn đến những cách hiểu không chính xác, những cách nhìn không hệ thống. Lấy một ví dụ đơn giản, nhiều sinh viên nghĩ rằng Turbo C, Borland C, Visual Studio C chính là C/C++ mà không hiểu bản chất của nó chỉ là 1 trình biên dịch của C/C++ có kèm theo trình soạn thảo.

   Để hiểu rõ hơn về bản chất của vấn đề chúng ta hãy làm quen với lập trình C trên Linux.

2. **Khởi đầu lập trình C với Linux**

   a. Những thao tác trước khi bắt đầu: Cài Linux và các gói development

   -- Kiểm tra sự tồn tại của các công cụ phát triển

   which gcc

   which make

   -- Kiểm tra công cụ soạn thảo: vim và gedit

   which vim

   which gedit

   b. Chương trình helloworld

   -- Dùng bất kì trình soạn thảo nào và đánh đoạn code sau vào. Save với tên là hello.c (ở đây dùng vim)

   vim hello.c

```
#include <stdio.h>
int main()
{
printf("Hello World \n");
return 0;
}
```

   --Biên Dịch:

   gcc -o hello hello.c

   -- Nếu không có lỗi gì thì sẽ ra file thực thi hello. Chạy thử bằng lệnh:

   ./hello

   -- Chú ý:

+ Dấu "." chính là thư mục hiện tại.
+ Linux dùng "/" thay vì "\" của DOS

3. Trình biên dịch GCC

- Trình biên dịch C trên Linux chỉ là một phần GCC (GNU Compiler Collection). GCC có thể dùng được với: C, C++, Objective C, Fortran,…
- Biên dịch C là "gcc"
- Lệnh xem version: *gcc --version*

4. Debug chương trình với GDB

GDB là một chương trình debug, gỡ rối chương trình rất phổ biến đối với công việc lập trình trong Linux.

Chúng ta sẽ đi tìm hiểu sơ lược về cách sử dụng chương trình. Giả sử ở đây, chúng ta có file date.c với nội dung như sau:

```
#include <stdio.h>
int main()
{
int day= 22;
int month= 12;
int year = 2014;
printf("\nHello all");
printf("\nChung ta se cung tim hieu ve GDB");
printf("\nNgay %d thang %d nam %d \n",day,month,year");
return 0;
}
```

Sử dụng gcc để biên dịch chương trình thành file thực thi, chú ý là các bạn phải thêm option [-g] để cho phép chạy debug.
Lệnh biên dịch như sau:
gcc –g –o date date.c

Sau khi compile xong, chúng ta sẽ được file thực thi **date**
Để bắt đầu debug file thực thi
gdb [ten_file_thuc_thi]
Ví dụ: *gdb date*

Trong console của GDB, ta có thể thực hiện các lệnh của linux
(gdb) shell [lenh_linux]
Ví dụ:
(gdb) shell clear

(gdb) shell ls

Để đặt break point ở một vị trí nào đó:

(gdb) break [so_dong]

hoặc

(gdb) break [ten_ham]

Ví dụ:

Cần đặt breakpoint tại dòng thứ 10 trong main.c

(gdb) break 10

hoặc đặt breakpoint tại hàm main

(gdb) break main

Để xóa breakpoint

(gdb) delete [so_thu_tu_break_point]

Ví dụ: Xóa breakpoint thứ 2

(gdb) delete 2

Sau khi đã đặt breakpoint(hoặc không cần đặt breakpoint tùy cách chúng ta debug), thì chúng ta bắt đầu chạy chương trình bằng lệnh sau

(gdb) run

Muốn xem toàn bộ chương trình:

(gdb) list

Ở ví dụ của chúng ta chỉ có hàm main, do đó khi dùng lệnh list thì sẽ hiển thị duy nhất các dòng code của hàm main. Còn nếu chương trình đang chạy ở 1 function nào đó, ví dụ function aaaa thì sẽ list code của function đó.

Chúng ta muốn chương trình chạy dòng lệnh kế tiếp thì trong gdb gõ

(gdb) next [so_dong]

Nếu không có [so_dong] thì dòng lệnh kế tiếp dòng lệnh hiện tại được thực thi, còn nếu có [so_dong] thì chương trình sẽ thực thi từ dòng lệnh hiện tại tới dòng lệnh [so_dong]

Muốn chạy vào bên trong thân hàm

(gdb) step

Muốn chương trình thực thi tiếp cho đến breakpoint kế hoặc đến hết chương trình

(gdb) continue

Trong quá trình chạy debug chúng ta muốn xem một biến có giá trị bằng bao nhiêu

(gdb) display [ten_bien]

Ví dụ: trong hàm main của chúng ta có biến day, mà muốn xem biến này chúng ta thực hiện lệnh

(gdb) display day

Muốn in giá trị của 1 biến trong console

(gdb) print [ten_bien]

Ví dụ: (gdb) print day

Muốn in địa chỉ của biến

(gdb) print &[ten_bien]

Ví dụ: (gdb) print &day

Hiển thị kiểu dữ liệu của biến

(gdb) ptype [ten_bien]

hoặc

(gdb) whatis [ten_bien]

Ví dụ

(gdb) ptype day

Gán giá trị cho 1 biến

(gdb) set variable [ten_bien] = [value]

(gdb) continue

Ví dụ:

(gdb) set variable day = 15

(gdb) continue