**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Viện Công Nghệ Thông Tin và Truyền Thông**

**-----o0o-----**

**BÁO CÁO BÀI TẬP LỚN MÔN LẬP TRÌNH HỆ NHÚNG THEO CHUẨN ITSS**

Đề tài: Thang máy chuyển hàng

|  |  |
| --- | --- |
| **Nhóm** | **: 05** |
| **Nguyễn Đình Hùng** | **: 20131879** |
| **Nguyễn Đình Tạo** | **: 20133415** |
| **Vũ Anh Sơn** | **: 20133371** |
| **Nguyễn Đình Chính** | **: 20111243** |
| **Giáo viên hướng dẫn** | **: Bành Thị Quỳnh Mai** |

**Hà Nội, 5/2017**

**MỤC LỤC**

[Phần 1. Mô tả về đề tài 3](#_bookmark0)

[Phần 2. Trình bày về Project 5](#_bookmark9)

* 1. [Cấu trúc](#_bookmark10) 5
  2. [Hoạt động](#_bookmark11) 5
  3. [Cài đặt và chạy](#_bookmark11) 12

Phần 3. Phân công nhóm 15

Phần 4. [Kết luận](#_bookmark39) 15

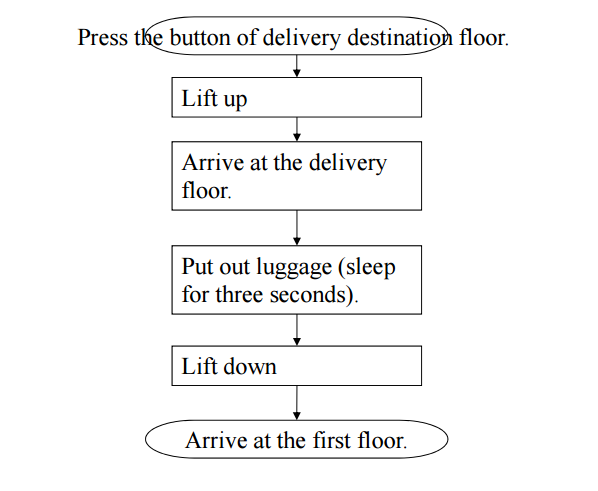
Phần 5. [Tài liệu tham khảo](#_bookmark40) 16

# PHẦN 1. MÔ TẢ ĐỀ TÀI

* Đề tài 2 : Luggage vehicular elevator control system Requirement specification
* Mô tả đề tài : Xây dựng hệt thống điều khiển thang máy vận chuyển hành lý ( từ giờ gọi tắt là thang máy) gồm có 5 tầng. Tất cả hành lý được tập trung ở tầng một, và các hành lý cần được chuyển lên các tầng 2-5 bằng thang máy. Ngoài ra thang máy còn có thể vận chuyển và thu hành lý từ các tầng 2-5 để xuống tầng một. Cụ thể :
* Viết chương trình theo dõi độ cao (vị trí) của thang máy và chương trình mô phỏng một bộ cảm biến của các tầng của thang máy, cụ thể tại mỗi tầng sẽ có các cảm biến ở trên cùng, đáy và ở giữa tầng. Các cảm biến sẽ là bật (turned) hoặc tắt (stop) dựa vào vị trí của thang máy. Nếu thang máy đang ở giữa tầng thì các cảm biến sẽ turned, còn nếu đi lên qua vị trí trên cùng hoặc xuống dưới qua đáy thì cảm biến sẽ chuyển sang stop.
* Hoạt động của hệ thống điều khiển thang máy :

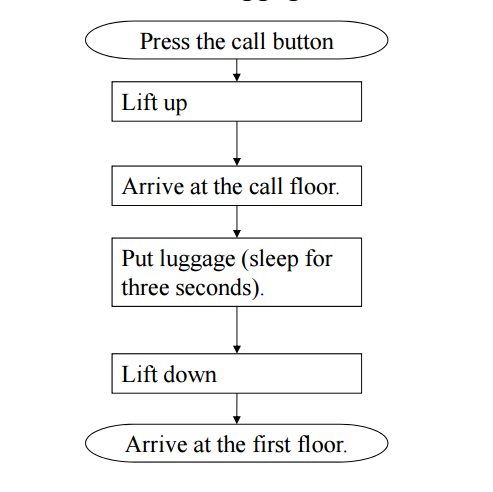
1. Hoạt động của tầng đầu tiên

Đưa hàng vào trong thang máy -> chọn tầng đích -> di chuyển -> giao hàng -> trở lại tầng một



1. Hoạt động của tầng 2-5th

Giả sử thang máy đang ở tầng một ( nếu không phải đợi thang máy về lại tầng 1), bầm nút gọi -> đưa hàng vào thang máy -> di chuyển lên tầng gọi -> giao hàng -> trở lại tầng một.



# PHẦN 2. TRÌNH BÀY VỀ PROJECT

## Môi trường & Ngôn ngữ

## Ngôn ngữ C.

## Linux (ubuntu).

## Git

## GTK+ 3 & C library.

## Cấu trúc và hoạt động của chương trình

1. **Cấu trúc**

Project bao gồm các file sau:

* **main.c**

Chứa hàm main để khởi chạy chương trình.

* **liftCtrl.c**

Chứa các hàm xử lí chính của chương trình : xác động vận tốc thang máy, mô phỏng sensor , xử lí các signal…

* **liftMng.c**

Nhận các signal request tới, đưa vào ngăn xếp hoặc gửi tới liftCtrl để xử lý request.

* **opepanel1.c**

Khởi tạo và xử lí giao diện của tâng 1 mỗi khi nhận được signal đồng thời gửi các đi các signal yêu cầu.

* **opePanelX.c**

Khởi tạo và xử lí giao diện các tâng 2, 3, 4, 5 mỗi khi nhận được signal đồng thời gửi các đi các signal yêu cầu.

* **sigs.c**

Chứa các hàm để cập nhật pid của vùng nhớ chia sẻ, giải phóng vùng nhớ chia sẻ và gửi signal.

* **sigs.h**

Định nghĩa (define) các signal và khai báo các hàm trong sigs.c

1. **Hoạt động**

## Sau khi hàm main ( trong main.c ) được chạy nó sẽ khởi tạo giao diện các tâng 1 đến 5, và các chương trình mng, ctr.

## 

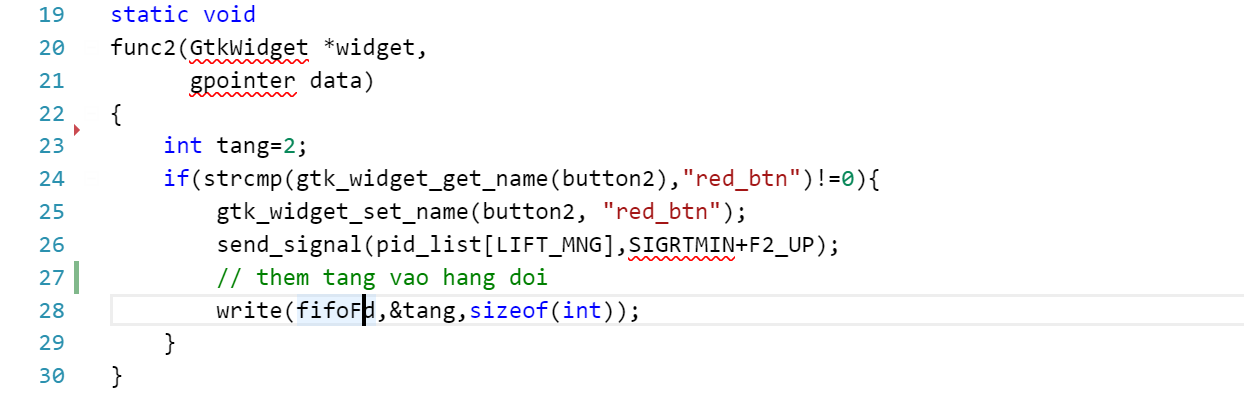
## 

*Hình 3. Hàm main*

## Khi một button của một tầng được ấn, chương trình opX (x là number của tầng đó) sẽ gửi signal yêu cầu tới chương trình mng.

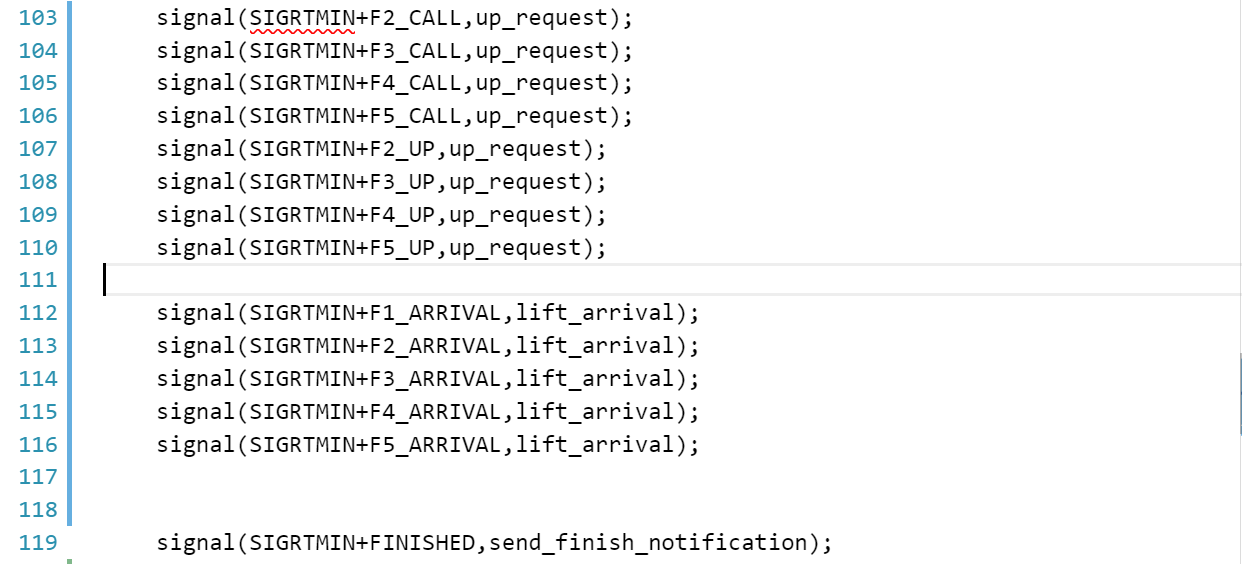
## 

## *Hình 4. Hàm call\_func trong opePanelX.c (xử lí khi nút call của các tầng 2-5 được ấn)*

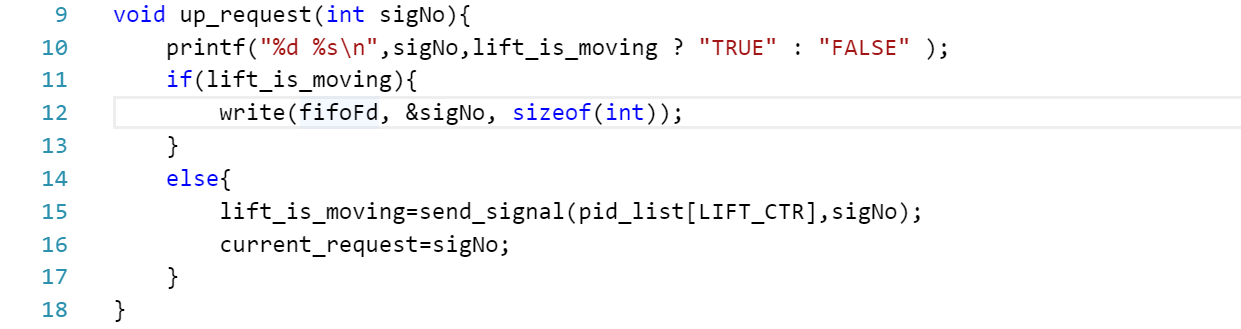


*Hình 5. Hàm func2 trong opePanel1.c (xử lí khi button 2 của tầng 1 bị ấn).*

Chương trình Mng nhận được signal sẽ xác định signal thuộc loại nào rồi sẽ kết hợp với trạng thái của thang máy (do sensor xác định) để chọn xử lí tương ứng : đưa vào hàng đợi hay lấy request mới từ hàng đợi ra hay gửi cho chương trình Ctr để xử lí ngay…



*Hình 6. Đoạn code nhận tín hiệu trong hàm main của liftMng.c*



*Hình 7. Hàm xử lí trong liftMng.c khi signal là up\_request.*



*Hình 8. Hàm xử lí trong liftMng.c khi signal là lift\_arrival*

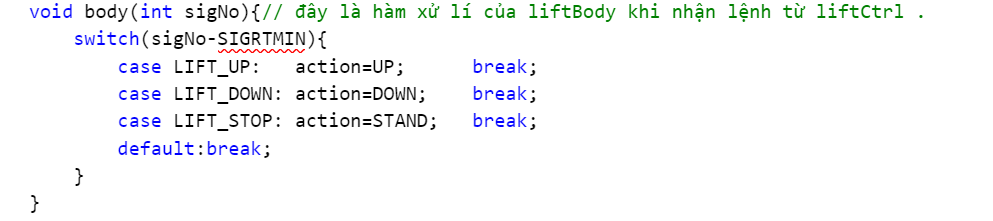
Một tiến trình con của Ctr khi nhận được signal từ chương trình Mng gửi tới sẽ thực hiện xử lí ứng với mỗi loại signal. Giả sử đối với tín hiệu up\_request

**

*Hình 9. Hàm up\_request trong liftCtr.c*

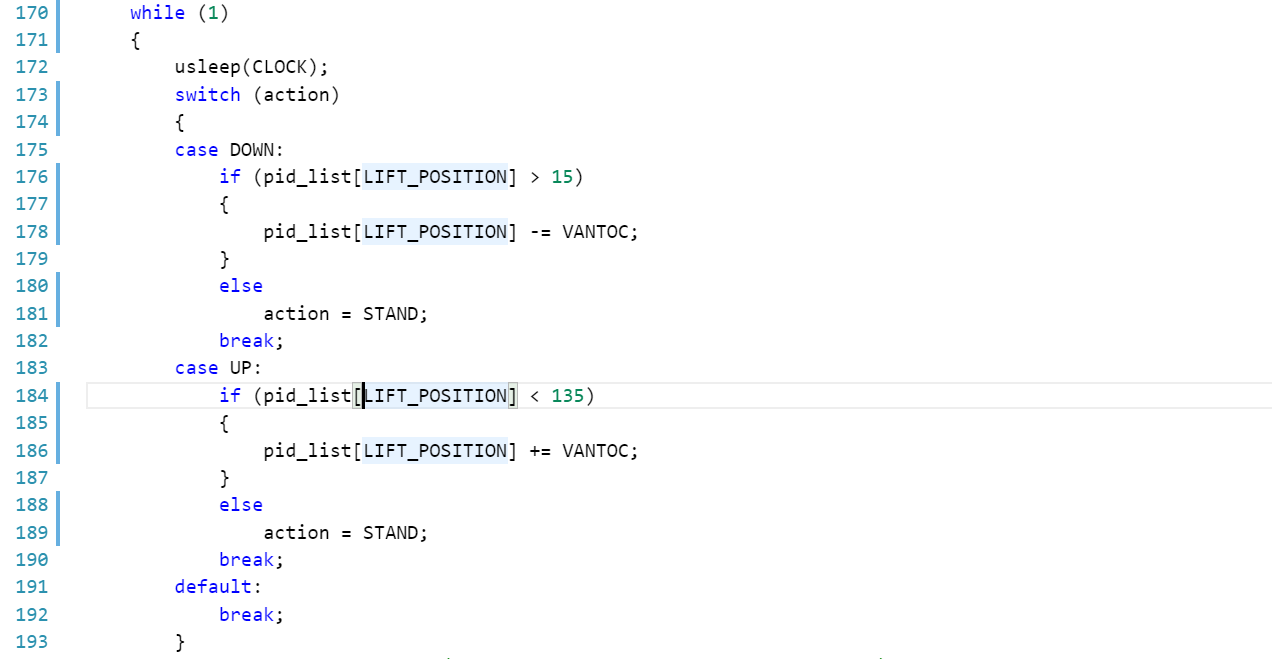
Tại đây, nó sẽ kill tiến trình con hiện tại và gửi tới tiến trình cha Ctr tín hiệu SIGRTMIN+LIFT\_UP.

Với tín hiệu đó,biến action (hành động mà thang máy sẽ thực hiện) sẽ được gán bằng UP.



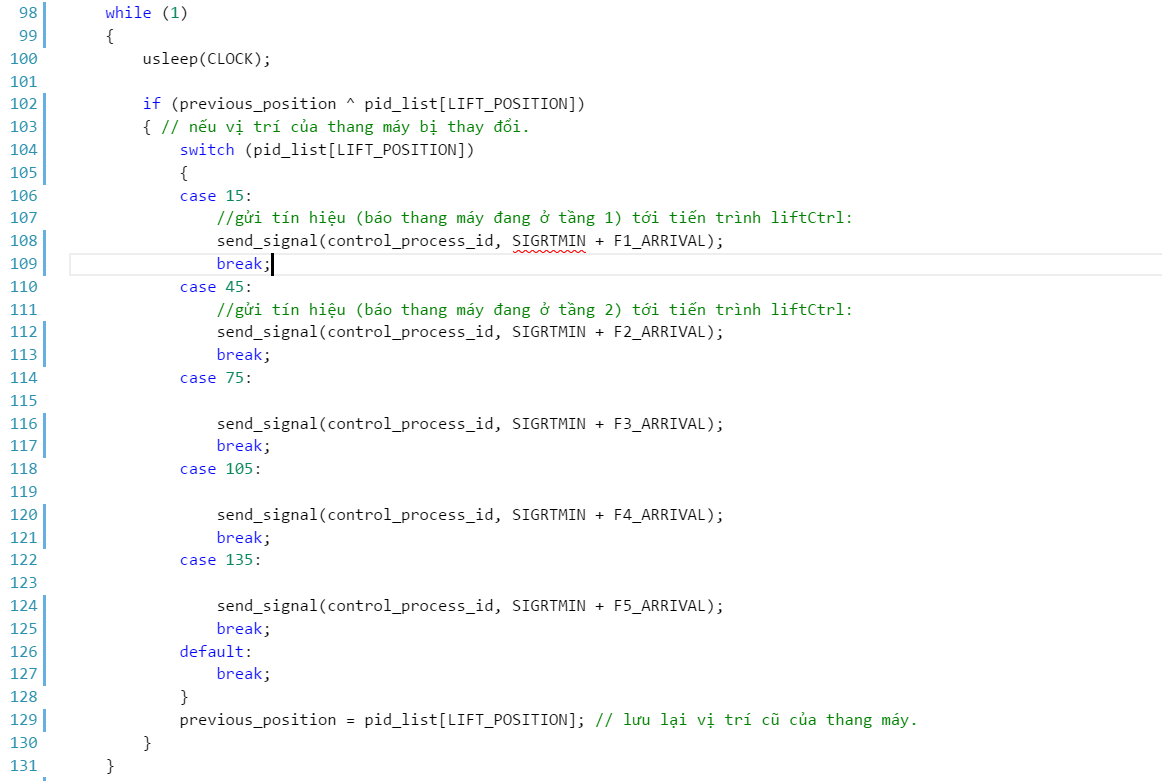
*Hình 10. Hàm body trong liftCtr.c set biến action.*

Khi action là UP thì chiều cao thang máy sẽ được tính dựa vào vận tốc theo đoạn mã bên dưới.



*Hình 11. Đoạn code trong liftCtr.c tính độ cao thang máy*

Độ cao của thang máy luôn được theo dõi bởi hàm sensor\_proccess\_run ()



*Hình 12. Đoạn code trong hàm sensor\_proccess\_run () theo dõi độ cao thang máy.*

Dựa vào độ cao sẽ xác định được thang máy đến tầng nào và chương trình sẽ gửi tín hiệu ARRIVAL báo đã đến tầng bao nhiêu cho chương trình Ctr.



*Hình 13. Hàm sensor\_change xác định thang máy đã đến đích chưa*

Khi nhận được tín hiệu ARRIVAL thì nó sẽ gọi hàm sensor\_change() để xác định xem thang máy đã tới đích chưa, nếu đã tới đích nó chuyển trạng thái sensor sang stop, và gửi signal FINISHED tới chương trình Mng (Mng sẽ chuyển tiếp chúng tới lại các chương trình Opx của các tầng thông báo đến trên giao diện) và signal LIFT\_DOWN cho tiến trình cha của chương trình Ctr để đưa thang máy đi xuống.

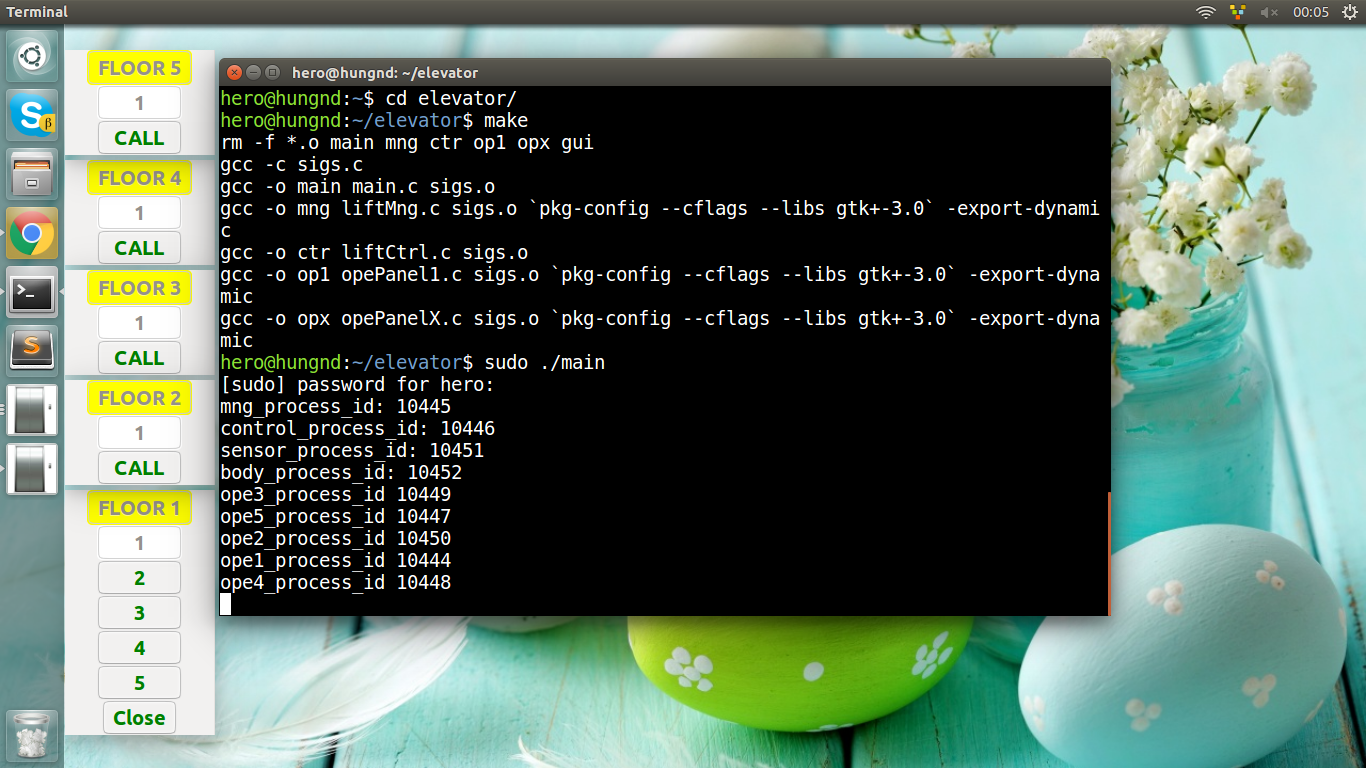
## Cài đặt và chạy chương trình

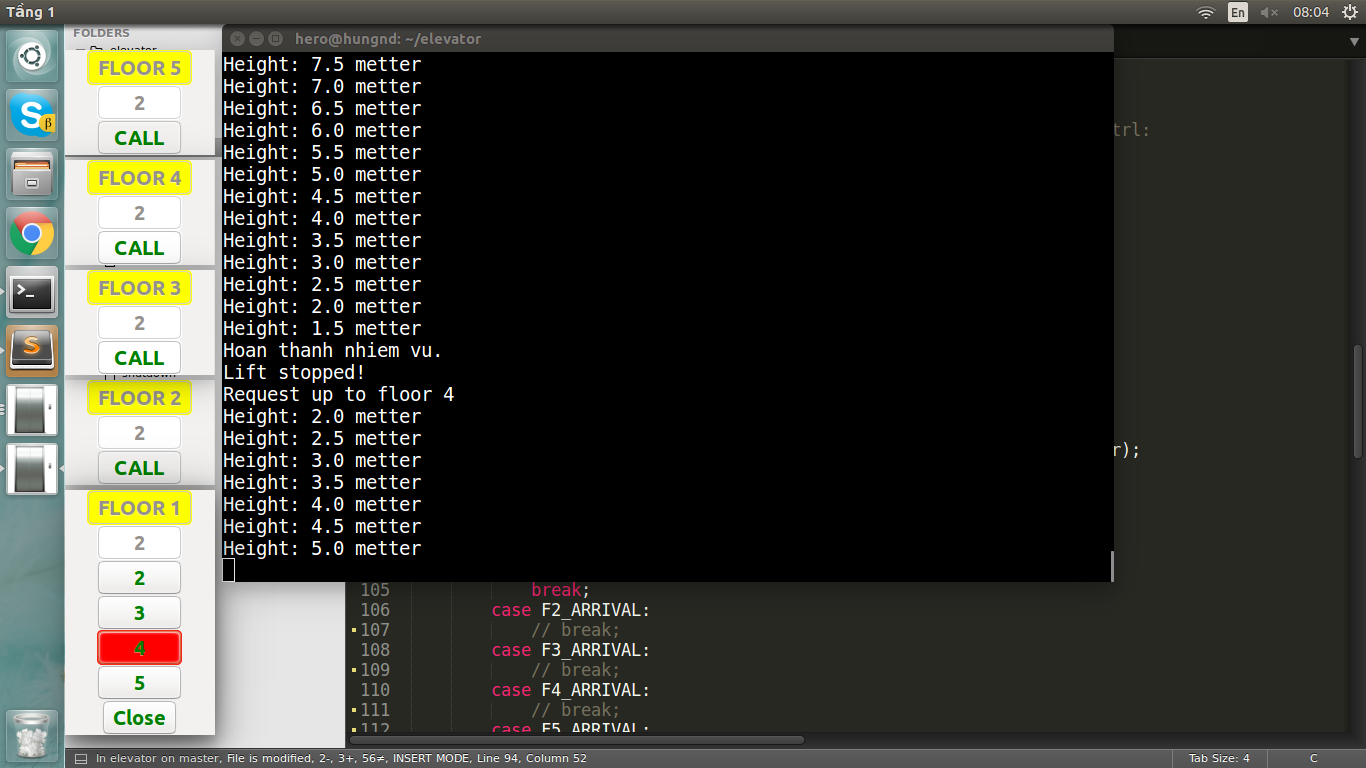
## Lấy source code: git clone https://github.com/huntojkojqw01/elevetor.git

## Chạy makefile

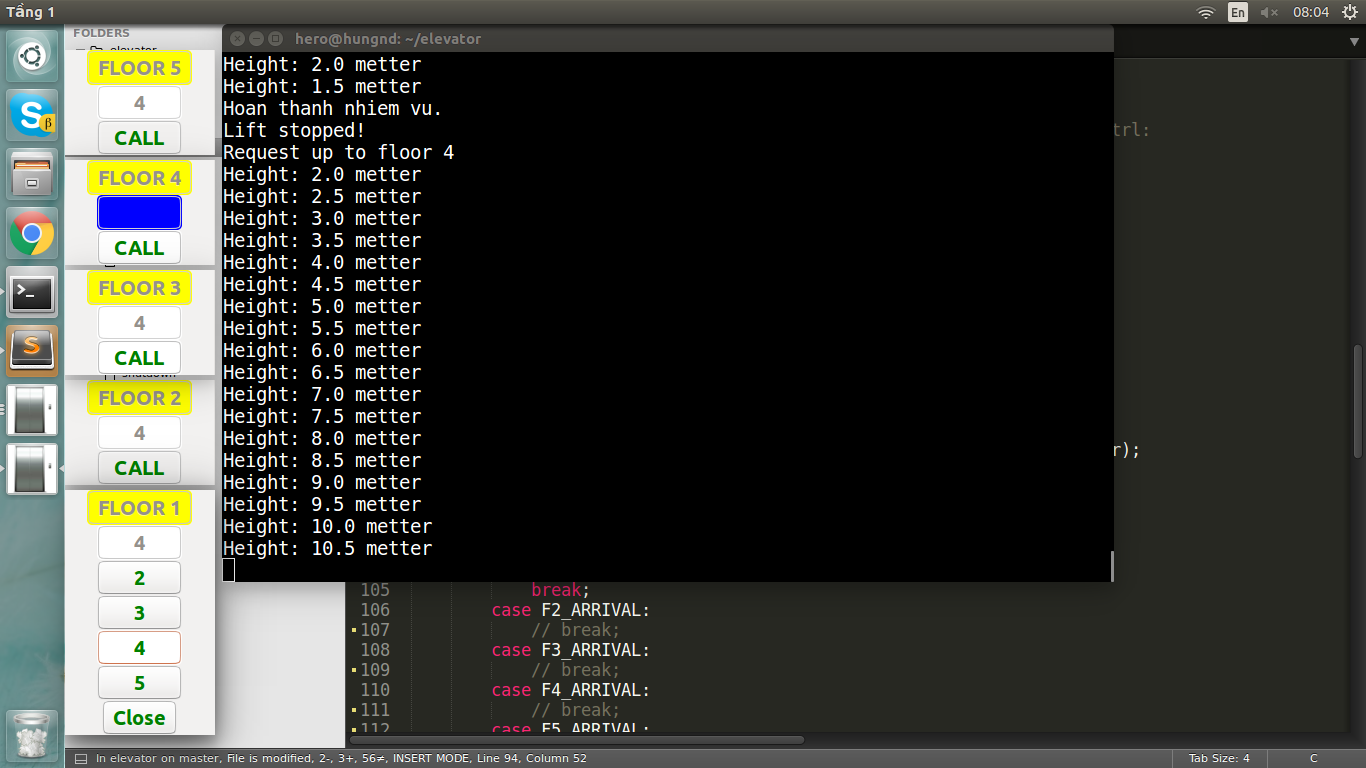
## Chạy sudo ./main

Dưới đây là một số hình demo chương trình :

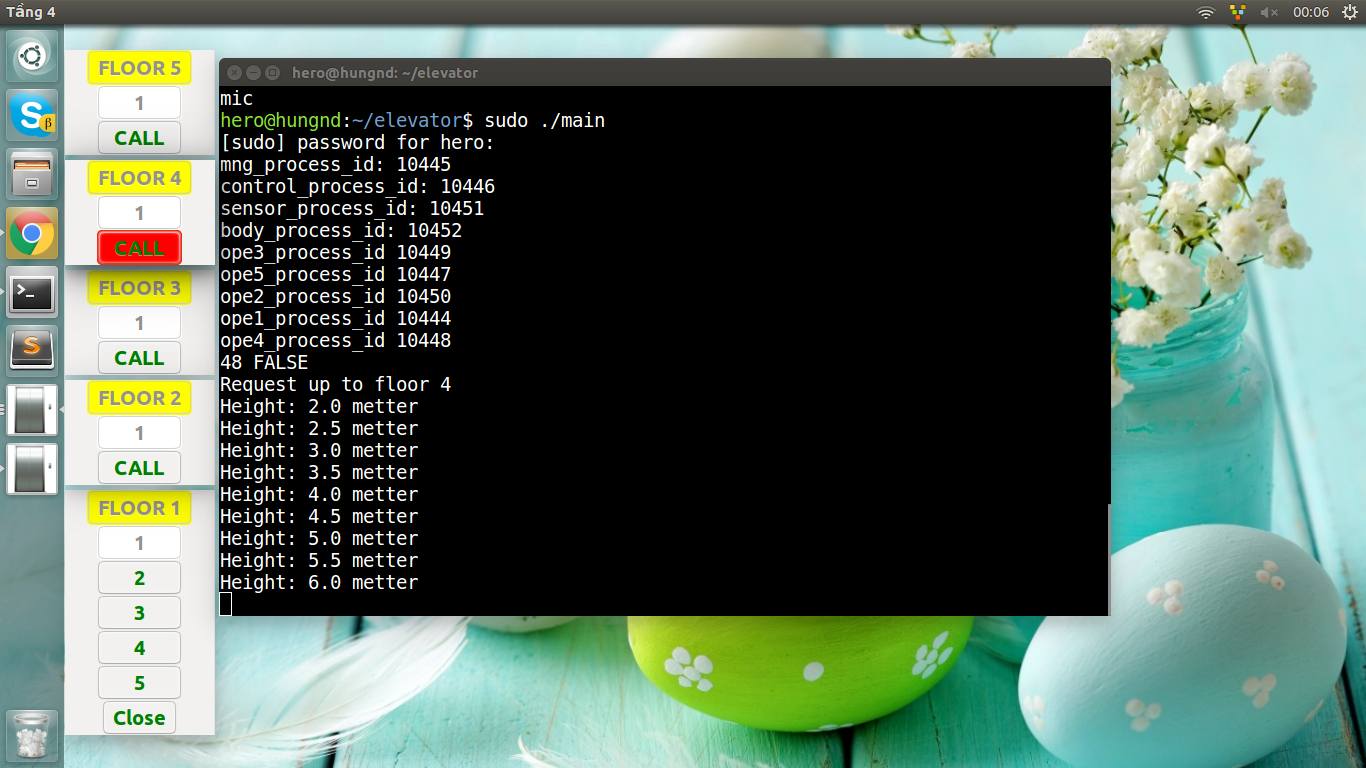


*Hình 14. Demo 1.*

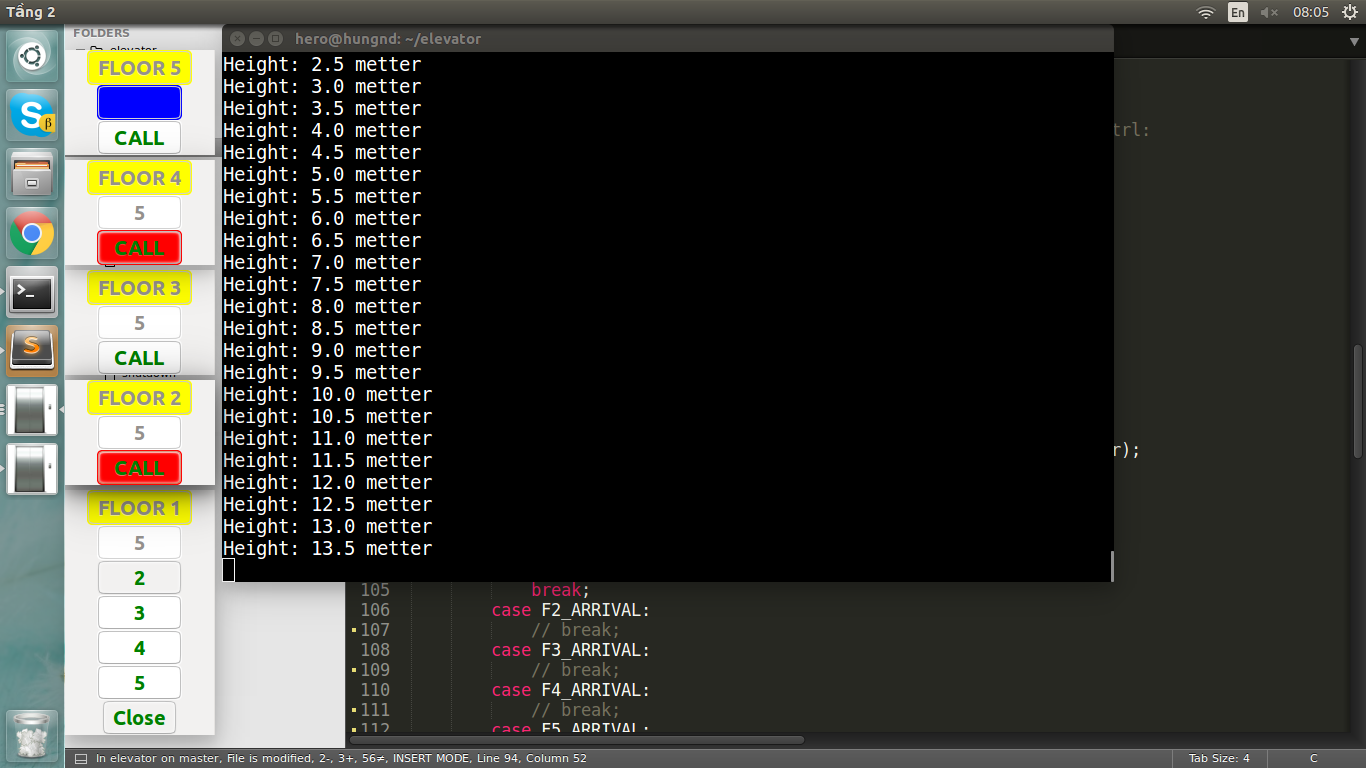
*Hình 15. Demo 2.*



*Hình 16. Demo 3*



*Hình 17. Demo 4*



## *Hình 18. Demo 5*

# Phần 3. PHÂN CÔNG NHÓM

# Nguyễn Đình Hùng (nhóm trưởng) phân chia công việc, hỗ trợ các thành viên , tổng hợp code. Nguyễn Đình Chính làm liftCtr.

# Nguyễn Đình Tạo làm opePanel,X và giao diện.

# Vũ Anh Sơn làm liftMng.

# Phần 4. KẾT LUẬN

# Chương trình của nhóm đã cơ bản hoàn thiện yêu cầu của đề tài đã chọn, và nhóm đã sử dụng GTK+3 để làm giao diện trực quan hơn. Qua quá trình tìm hiểu và làm project này, nhóm đã hiểu và thực hành được các lí thuyết đã học trên lớp ( MultiProcCtr, SharedMem, Fifo, SignalH … ).

# TÀI LIỆU THAM KHẢO

[1]. Slide môn lập trình hệ nhúng theo chuẩn ITSS.