# Câu 2:

## Đề bài

Xây dựng hệ thống chương trình cho phép liên lạc giữa các tiến trình trong 1 máy và /hoặc trên các máy (với các hệ điều hành có thể khác nhau); sao cho chúng có khả năng phòng Deadlock hoặc tránh cùng rơi vào Critical section.

## Mô tả

Hệ thống giám sát /can thiệp thời gian sử dụng máy tính của trẻ em bao gồm 2 chương trình chính:

**Chương trình C (for Children):** chạy trên máy laptop /desktop hệ điều hành Windows của trẻ và được đặt ở chế độ Autorun (tự động chạy khi bật máy), thực hiện các việc:

1. Lấy mật khẩu (từ bàn phím)
2. Nếu chuỗi nhập là mật khẩu của phụ huynh: Chương trình đợi đến 60 phút sau thì mới quay lại hỏi lại mật khẩu (thực hiện lại bước C0) // lúc này là phụ huynh sử dụng máy chứ không phải trẻ!
3. Nếu không phải là mật khẩu của phụ huynh:
   1. Kiểm tra xem thời điểm hiện tại có nằm trong khung thời gian trẻ chưa được dùng máy hay không:
      1. *Nếu đang trong khoảng thời gian trẻ chưa được dùng máy:*

Thông báo tới khi nào mới được dùng (hiển thị ra màn hình và /hoặc nói ra loa), sau đó thực hiện song song 2 việc: (1): kiểm tra xem đã đủ 15 giây chưa kể từ lúc thông báo xong & nếu đã đủ thì chương trình tự tắt máy (shutdown hệ điều hành -không cho người dùng can thiệp) – (2) thực hiện lại từ đầu việc C0 & C1 (tức nếu người dùng kịp nhập đúng mật khẩu phụ huynh thì không tắt máy mà thực hiện C1 - đợi đến 6 phút sau…)

* + 1. *Ngược lại (đang trong khoảng thời gian trẻ được dùng máy):* 
       1. *Nếu mật khẩu không phải mật khẩu của trẻ:* thực hiện lại việc hỏi và kiểm tra mật khẩu (thực hiện lại C0) cho đến lần nhập sai mật khẩu thứ 3 thì đặt thời gian không được dùng máy là 10 phút kể từ thời điểm hiện tại rồi tắt máy.
       2. *Ngược lại (đúng mật khẩu của trẻ):* Đọc thông tin về khung giờ được dùng (a) và Thông báo còn bao nhiêu phút nữa máy sẽ, sau đó chạy ở chế độ giám sát thực hiện cùng lúc các việc: (1) Sau mỗi 0.13 giây lưu lại nội dung các phím đã gõ, (2) thực hiện (a) và thấy có thay đổi (do cha /mẹ chạy tiến trình P và điều chỉnh) thì cập nhật thông tin và thông báo còn mấy phút nữa tắt máy, (3) kiểm tra thấy còn 1 phút đến thời điểm tắt máy thì thông báo và còn 0 phút thì tắt máy. Thông tin về khung giờ sử dụng sẽ được cập nhật

Thông tin sẽ được lưu trên cloud Firebase sẽ có hai loại là Schedule cho phần lịch cho phép dùng máy và History là lịch sử sử dụng máy của trẻ.

Trong Schedule: mỗi lịch sẽ có các trường: date dùng để thể hiện ngày, duration dùng để thể hiện thời gian tối đa 1 lần máy có thể được mở(tính theo phút), interruptTime thể hiện thời gian ngắt(tính theo phút), sum thể hiện thời gian tổng trong 1 lần mở máy, timeStart thể hiện thời gian bắt đầu có thể mở máy (HH:mm), timeEnd thể hiện thời gian kết thúc có thể mở máy(HH:mm)

Ví dụ cụ thể, với nội dung Schedule như sau: TimeStart là 14:20 đến timeEnd là 17:20, duration bằng 60, interruptTime bằng 12 sum bằng 126 thì các khung giờ được dùng là: Trong khoảng thời gian từ 07:30 đến 11:30 có thể sử dụng máy, nhưng mỗi lần bật máy thì chỉ được dùng tối đa 60 phút – sau đó máy sẽ không hoạt động cho đến khi đã ngắt đủ 12 phút, đồng thời khi đã dùng đủ 126 phút thì máy cũng sẽ không chịu chạy nữa.

Trong History: Mỗi lịch sử sẽ có các trường: date để thể hiện ngày, keyLog để thể hiện các phím được nhấn trong lần sử dụng đó, timeStart để thể hiện thời gian mở máy, timeEnd để thể hiện thời gian tắt máy.

Ví dụ cụ thể, với nội dung History như sau: TimeStart là 14:20 đến timeEnd là 17:20, keyLog lưu chuỗi ký tự mà : Trong khoảng thời gian từ 07:30 đến 11:30 có thể sử dụng máy, nhưng mỗi lần bật máy thì chỉ được dùng tối đa 60 phút – sau đó máy sẽ không hoạt động cho đến khi đã ngắt đủ 12 phút, đồng thời khi đã dùng đủ 126 phút thì máy cũng sẽ không chịu chạy nữa.

**Chương trình P (for Parent):** Chạy trên điện thoại hệ điều hành Android, thực hiện các việc:

1. Lấy mật khẩu (từ bàn phím)
2. *Nếu chuỗi nhập không phải là mật khẩu của phụ huynh:* Chương trình xuất ra thông báo sai mật khẩu rồi cho phép nhập lại
3. *Nếu chuỗi nhập là mật khẩu của phụ huynh:* Vào giao diện chính của chương trình, những lần sau chỉ cần mở ứng dụng không cần đăng nhập lại, nếu muốn đăng nhập lại phải đăng xuất chương trình
   1. *Trong giao diện chính của chương trình có 2 tab lịch sử thể hiện lịch sử sử dụng máy và lịch thể hiện các mốc thời gian có thể sử dụng máy:*
      1. *Tab lịch:*
         1. Hỏi người dùng ngày muốn xem lịch sau khi nhập ngày thì có thể thêm lịch, xóa lịch, sửa lịch, xem chi tiết lịch. Từng lịch thể hiện ngày, thời gian bắt đầu và kết thúc
      2. *Tab lịch sử:*
         1. Hỏi người dùng ngày muốn xem lịch sử sau khi nhập ngày thì cho phép người dùng xem lịch sử mà trẻ mở máy vào các ngày (ngày mở, thời gian bắt đầu, thời gian kết thúc). Từng lịch sử có thể cho người dùng thông tin bàn phím nhập trong lúc đó. Nếu lịch sử có giờ kết thúc bằng -1 thì tức là trẻ đang online).

Lưu ý:

## Cách giải quyết Deadlock và Critical Section (Mã giả)

* **Cách giải quyết:**
* Sử dụng cờ hiệu
* **Mã giả (Đoạn code bị miền găng)**

if(Các dữ liệu hợp lệ)  
 if(trường hợp thêm lịch mới){  
 Schedule schedule\_final 🡪 new Schedule(dữ liệu nhập và cho biến cờ hiệu ban đầu là “0”);

//add dữ liệu vào Cloud  
 scheduleDBM.add(schedule\_final).addOnSuccessListener(suc ->  
 {  
 Thêm thành công thì hiện thông báo  
 }).addOnFailureListener(er ->  
 {  
 Thêm thất bại thì hiện lỗi  
 });  
 }  
 else {//trường hợp chỉnh sửa lịch  
 HashMap<String,Object> hashMap🡪new HashMap<>();  
 hashMap 🡪 dữ liệu chỉnh sửa mới+ biến cờ hiệu là “1”;  
 scheduleDBM.getFlag(schedule.getKey()).addOnSuccessListener(new OnSuccessListener<DataSnapshot>() {//thành công xin cờ hiệu ở lịch mà mình muốn sửa  
 @Override  
 public void onSuccess(DataSnapshot dataSnapshot) {   
 if (cờ hiệu lấy được == “0”) {  
 db.update(hashMap)// update dữ liệu cũ bằng dữ liệu mới

Hiện thông báo “Cập nhật thành công”

} else {//cờ hiệu lấy được ==”1”  
 Hiện thông báo “Dữ liệu đang được truy xuất”  
 }  
 }  
 }).addOnFailureListener(new OnFailureListener() {//Thất bại khi xin cờ hiệu mà mình muốn sửa  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Hiện thông báo lỗi  
 }  
 });  
 }  
 Quay trở lại màn hình Lịch  
}

* **Ưu điểm:**
* Code dễ hiểu, dễ dàng chỉnh sửa và phát triển
* Giải quyết được
* **Khuyết điểm:**
* Không đảm bảo không có 2 tiến trình nào ở cùng miền găng cùng lúc (mutual exclusion) do code kiểm tra cờ hiệu và dành quyền cũng nằm trong miền găng
* **Hướng phát triển thuật toán:**
* Sử dụng method runTransaction(…) được hỗ trợ bởi Firebase để chịu trách nhiệm lắng nghe dữ liệu và trả về Transaction.Result nếu dữ liệu mới tại vị trí muốn thay đổi nên được tiến hành hay hủy bỏ.

## Video demo

* Link youtube:
* Link drive (dự phòng):

## Tài liệu tham khảo

* https://stackoverflow.com/questions/60775018/getting-data-from-firebase-in-a-synchronized-way
* https://stackoverflow.com/questions/47847694/how-to-return-datasnapshot-value-as-a-result-of-a-method/47853774
* https://firebase.google.com/docs/reference/android/com/google/firebase/database/Transaction.Handler