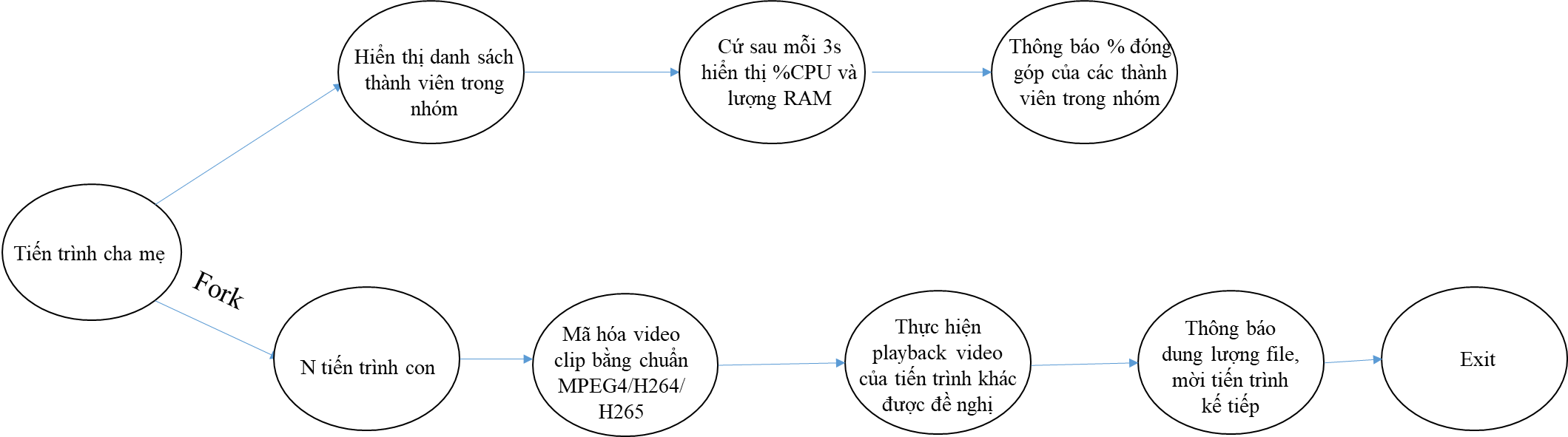
Chương 3: Thiết kế

* 1. Mô hình tổng quát thiết kế chương trình



Hình 3.1. Mô hình tổng quát chương trình

* + 1. Tiến trình cha mẹ

Tiến trình cha mẹ là tiến trình được sinh ra ngay sau khi chương trình được chạy (là hàm main() trong lập trình c). Trong thiết kế của chương trình tiến trình cha mẹ sau khi được sinh ra sẽ thực hiện lần lượt các công việc:

* Thực hiện hiển thị tên của các thành viên trong nhóm
* Sinh ra 5 tiến trình con
* Cứ sau mỗi 3s in ra màn hình %CPU và dung lượng RAM mà tiến trình chiếm đóng
* Chờ các tiến trình con thực hiện xong thì thông báo ra màn hình phần trăm đóng góp của các thành viên trong nhóm
* Kết thúc chương trình
  + - 1. Hiển thị danh sách các thành viên trong nhóm

Chương trình được viết trong bài tập lớn sử dụng ngôn ngữ lập trình c. Phần chương trình thực hiện n ra màn hình danh sách các thành viên trong nhóm:

printf("Thanh vien trong nhom \n");

printf("1. Hoang Van Toan - 20153825 \n");

printf("2. \n");

printf("3. \n");

printf("4. \n");

printf("5. \n");

Phần chương trình sử dụng lệnh printf trong c được cung cấp bởi thư viện stdio.h trong c.

* + - 1. Sinh ra 5 tiến trình con

Ở phần này, có hai công việc cần thực hiện là tạo ra một tiến trình trong c sau đó dùng lập trình đa luồng trong C thực hiện công việc cho 5 tiến trình con này được chạy gần như là cùng một lúc (song song với nhau). Để sinh ra một tiến trình con trong c, ta sử dụng lệnh fork(). Khi ta gọi lệnh fork() thì chương trình sẽ tạo ra một tiến trình con. Phần code c thực hiện sinh ra một tiến trình con:

fork(); // khởi tạo tiến trình con

wait(NULL); // tiến trình chính không chờ đợi gì ở tiến trình con

\*/ do something \*/ // Các công việc mà tiến trình con thực hiện

exit(0); // Kết thúc tiến trình con

Khi lệnh fork() được gọi, chương trình chính sẽ yêu cầu hệ điều hành tạo ra một tiến trình con có tất cả các trạng thái có thể có như các tiến trình khác.

Sau khi thực hiện tạo ra các chương trình con bằng lệnh fork(), chúng ra cần cho các tiến trình này chạy song song với nhau bằng lập trình đa tuyến trong c.

* + - 1. Cứ sau mỗi 3s in ra màn hình phần tram CPU và dung lượng RAM
      2. In ra màn hình phần trăm đóng góp của các thành viên trong nhóm

Tương tự như phần in ra màn hình danh sách các thành viên trong nhóm, ta cũng sử dụng lệnh printf() trong c để in dữ liệu ra màn hình.

* + - 1. Kết thúc chương trình

Để kết thúc tiến trình cha mẹ trong c, trong phần chương trình main ta sử dụng lện return 0 để kết thục tiến trình trong c. Khi gặp lênh return thì chương trình sẽ được kết thúc và giải phóng các tài nguyên cho hệ điều hành.

* + 1. Tiến trình con

Chương trình chính khi chạy sinh ra 5 tiến trình con, mỗi tiến trình con thực hiện các công việc:

- Mã hóa video clip bằng chuẩn MPEG4/H264/H265

- Thực hiện phát lại video của tiến trình khác khi nhận được yêu cầu của tiến trình đó

- Thông báo ra màn hình dung lượng file, chiều dài thời gian video

- Mời tiến trình con kế tiếp thực hiện

- Kết thúc tiến trình

3.1.2.1. Mã hóa video clip bằng chuẩn MPEG4/H264/H265

Để thực hiện mã hóa video clip trong c, chúng ta gọi lệnh system(“lệnh trong Ubuntu”). Sauk hi gọi lệnh system thì chương trình sẽ thực hiện gọi lệnh trong Ubuntu và thực hiện câu lệnh đó. Phần code thực hiện mã hóa video clip:

system("ffmpeg –i ‘path video input’ ‘path video output’");

Trong câu lệnh trên sử dụng lệnh trong Ubuntu ffmpeg –i ‘path video input’ ‘path video output’ thực hiện mã hóa video với đường link là path video input với chuẩn của video input thành video output với đường link cuẩ video output được viết trong path video output với chuẩn mà mình quy định.

3.1.2.2. Phát video clip

Tương tự như phần mã hóa video clip ở trên, ở phần này cũng sử dụng lệnh của Ubuntu và sử dụng một chương trinh để phát lại video là Vlc. Phần chương trình phát video trong chương trình c:

system("vlc ‘path video’");

3.1.2.3. Thông báo ra màn hình dung lượng file

3.1.2.4. Mời chương trình kế tiếp thực hiện

Ở đây, với các chương trình con của bài tập lớn đều được viêt bằng một hàm con trong c, nên khi mời chương trình con tiếp theo chạy thì trong chương trình con hiện tại chỉ cần gọi lại chương trình con tiếp theo thực hiện.

3.1.2.5. Kết thúc chương trình con

Tương tự như kết thúc tiến trình cha mẹ, khi gặp lệnh return trong c thì chương trình con cũng kết thúc và giải phóng tài nguyên lại cho hệ điều hành quản lí.