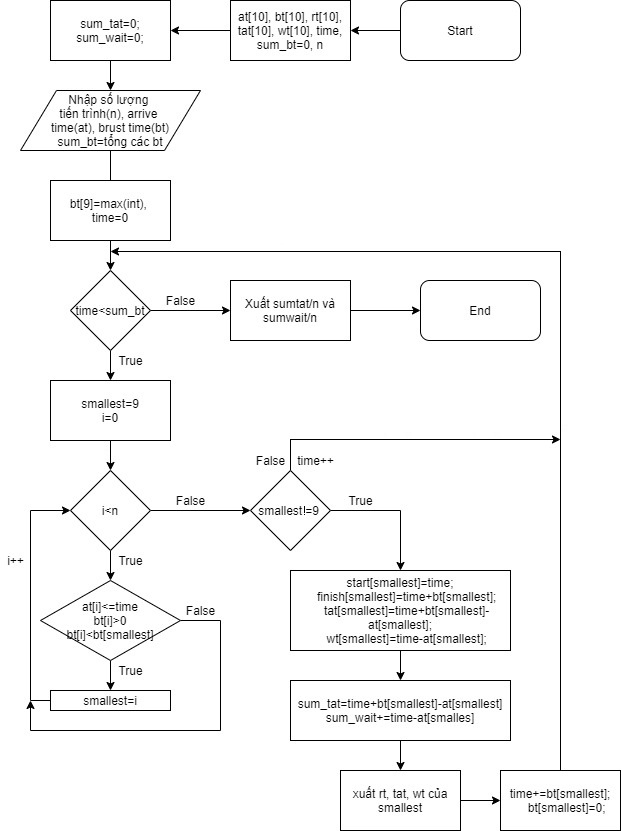
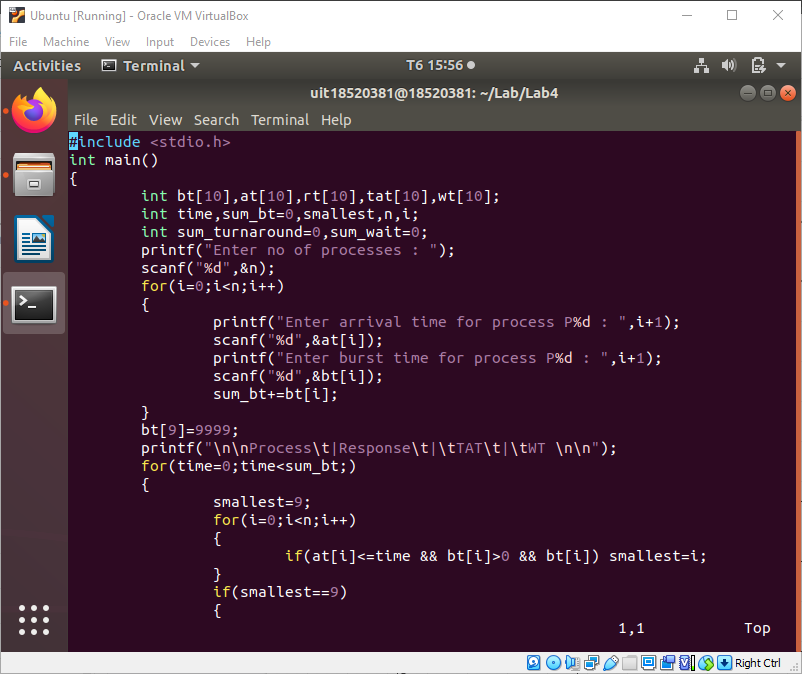
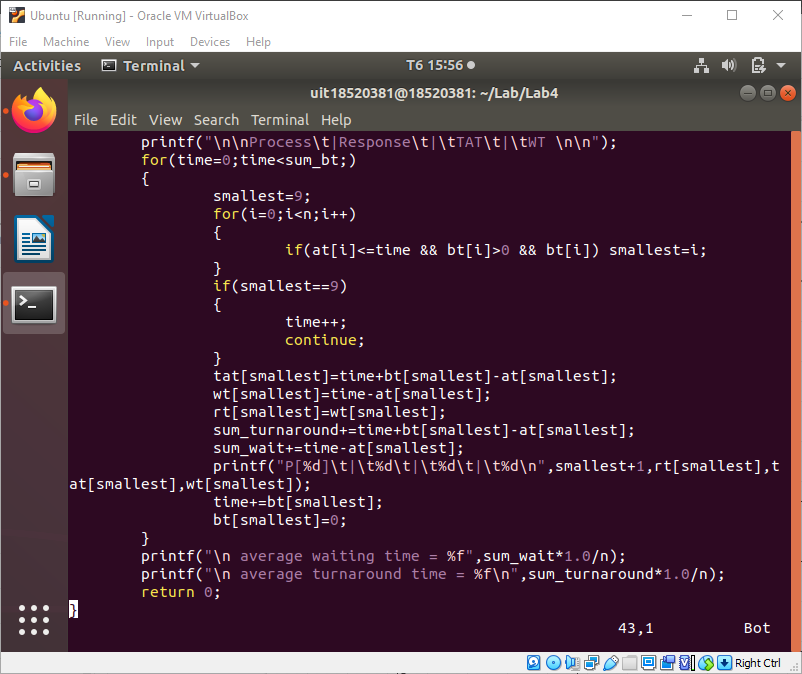
Sơ đồ thuật toán SJF: 

Ý tưởng:

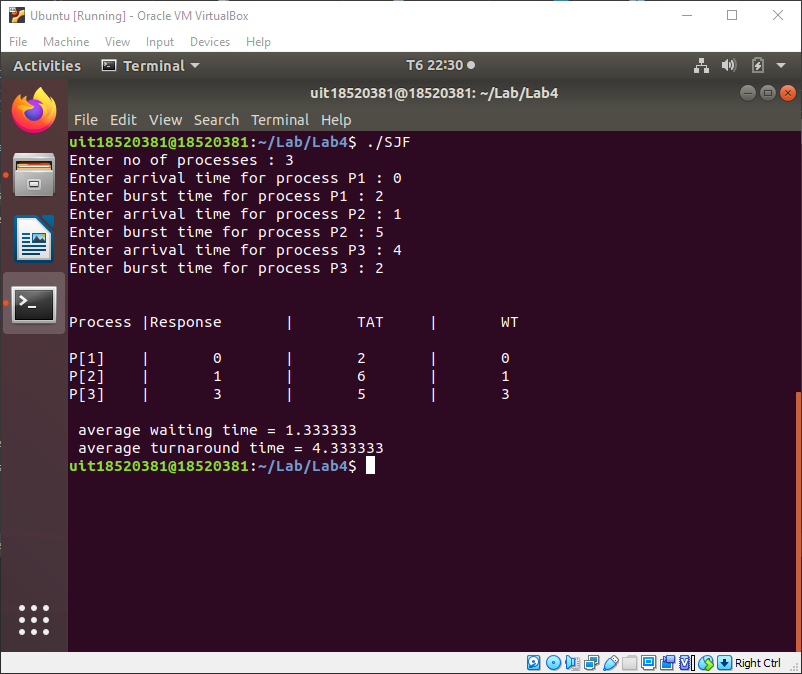
- Dùng 2 vòng lặp: 1 để giữ time cùng lúc đó tìm tiến trình được cho phép chạy, 1 vòng lặp để tìm tiến trình có burst time nhỏ nhất để chạy và gán nó là smallest, nếu không tìm được thì tiếp tục tăng time lên đến khi tìm được tiến trình phù hợp tiếp theo. Khi đã tim được thì thực hiện tính toán và gán cho các giá trị rt, tat, wt, cho smallest và trả burst time=0 để ngăn cho chương trình tìm và thực hiện lại tiến trình này. Khi đó ta tăng time lên bằng burst time của tiến trình smallest đó

- Sau khi kết thúc vòng lặp thì trả kết quả của sum\_tat trung bình và sum\_wait trung bình

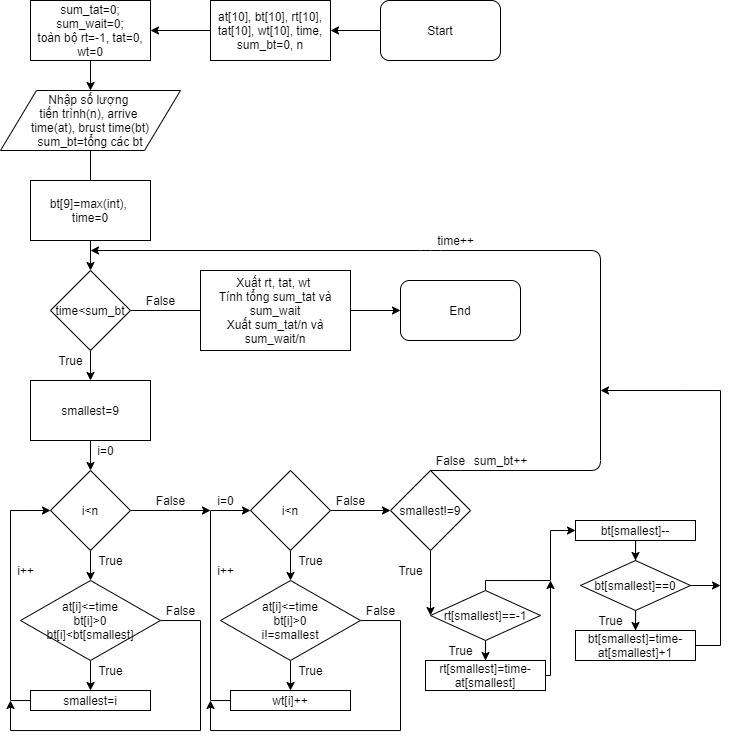
Code c:



Chạy mô phỏng với 3 process



Sơ đồ thuật toán STR:



Ý tưởng:

- Cũng giống như thuật toán SJF, nhưng lần này time sẽ chạy từng giây 1. Ban đầu ta khởi tạo các giá trị =0, riêng rt được gán =-1 để chỉ nhận giá trị 1 lần duy nhất

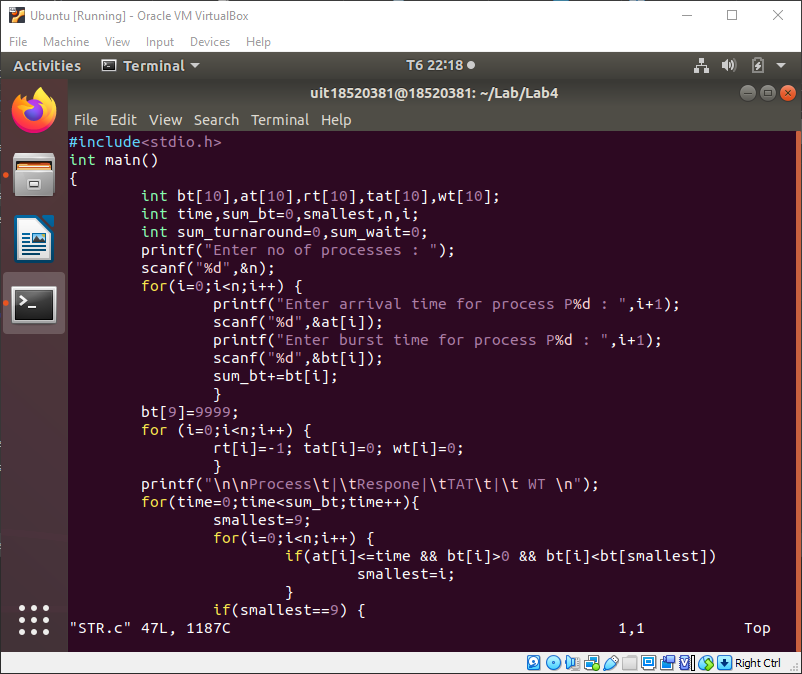
- Khi trong vòng lặp của time, ta vẫn tìm burst time nhỏ nhất và gán vào smallest. Nếu rt chưa nhận giá trị (=-1) thì ta tính kết quả gán vào, giảm burst time của smallest đi

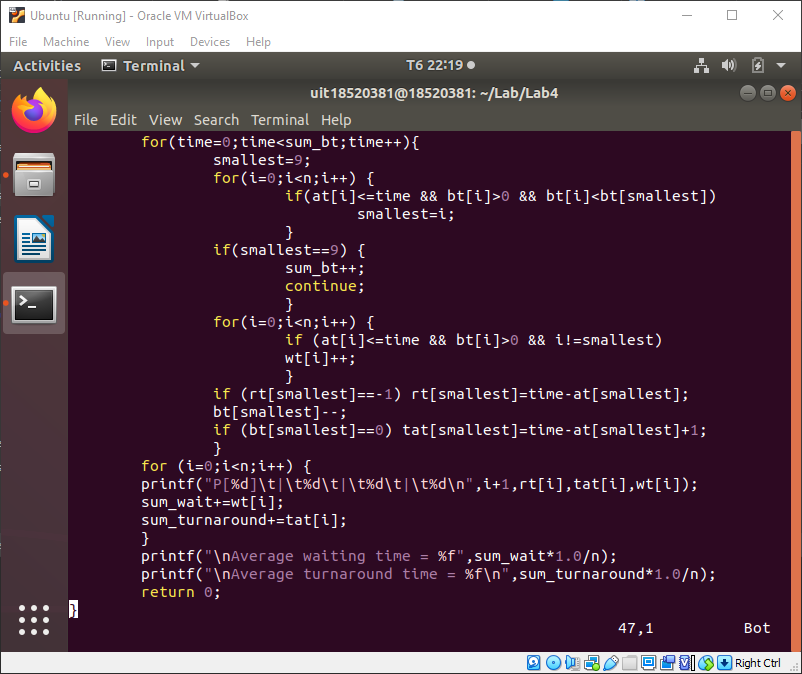
- Nếu burst time=0 (process kết thúc) thì ta tính tat

- Những process nào đã được đưa vào và chưa kết thúc và không phải smallest thì tăng wt lên 1

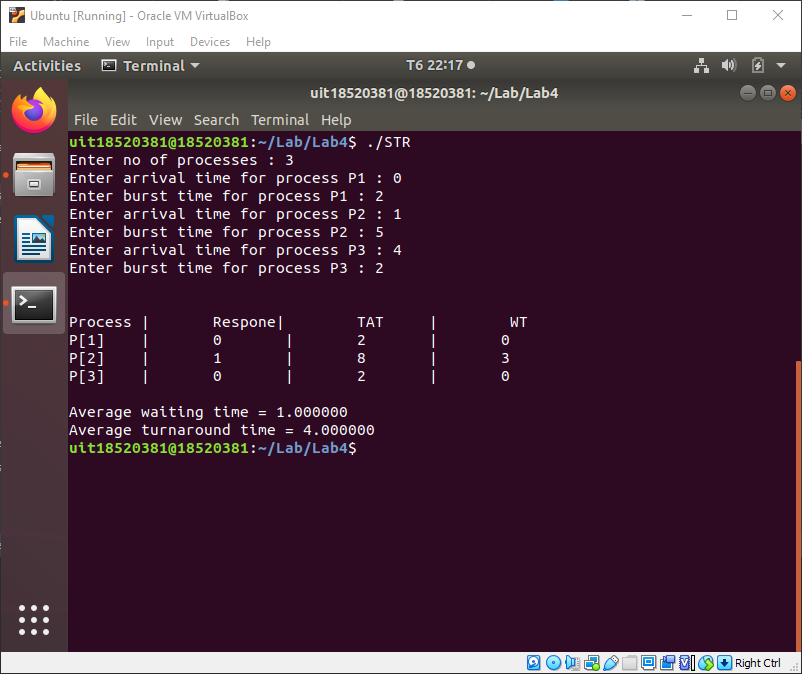
- Cuối cùng ta xuất toàn bộ rt, tat, wt, rồi tính tổng wt và tat để xuất ra thêm các giá trị trung bình cần tìm

Code c:





Chạy mô phỏng với 3 process



Sơ đồ thuật toán RR: (do trong lúc quét virus thì máy em quét luôn cả file của oracle nên Ubuntu bị lỗi nên em compiler đỡ bằng code:block)

Ý tưởng:

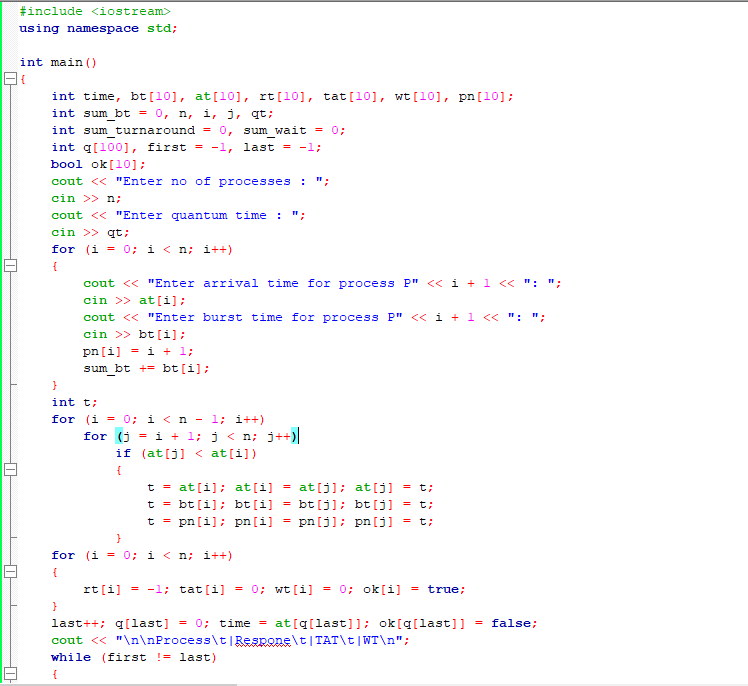
- Sử dụng 1 queue để chứa các process để đưa vào, sau đó sort theo thứ tự tăng dần arrive time(at), rồi đưa process đầu tiên vào hàng đợi.

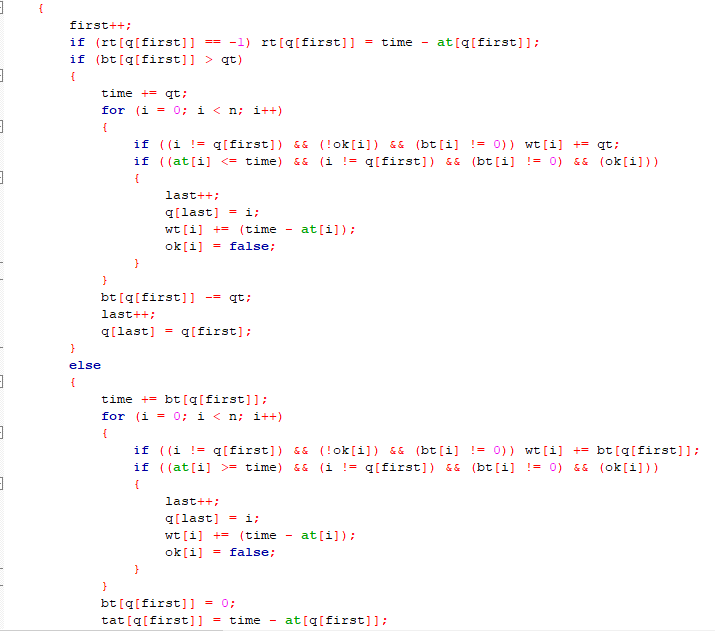
- Nếu trong hàng đợi còn thì ta vẫn chạy, response time(rt) =-1 của process đang xét trong queue thì ta trả về bằng thời gian hiện tại (time)-at của process đó, rồi xét process đó kết thúc chưa

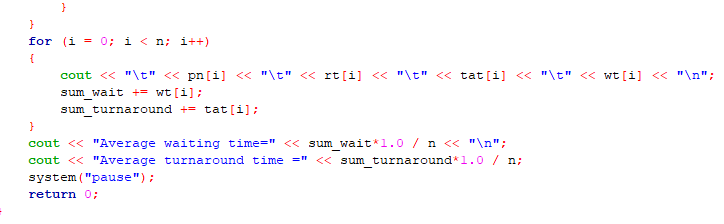
+ Nếu kết thúc (brust time(bt)<=quantum time(qt)), tăng time lên bằng bt của process đang xét, xét xem trong toàn bộ process còn lại nếu at[i]<=time và đã có trong hàng chờ (ok[i]) chưa, nếu chưa thì đưa vào trong queue; và xét nếu đã trong hàng chờ, chưa kết thúc (bt[i]>0), và khác tiến trình đang chạy thì tăng wait time (wt[i]) 1 khoảng bằng bt của process đang xét. Cuối cùng trả bt=0 và tính turn around time (tat)=time-at process đang xét

+ Nếu chưa kết thúc (bt>qt), thì ta cũng thực hiện như chưa kết thúc, chỉ khác ở bt đang xét bằng qt, không tính tat và thêm process đang thực hiện vào cuối queue

- Cuối cùng ta xuất toàn bộ process name (pn),rt, tat, wt, rồi tính tổng wt và tat để xuất ra thêm các giá trị trung bình cần tìm







Chạy mô phỏng với 3 process:

