Table of Contents

[**FCFSsaiLinked File 7**](#_Toc515252785)

[**SJFsaiLinked File 8**](#_Toc515252786)

[**LJFsaiLinked File 8**](#_Toc515252787)

[**SJFsaiLinked File 9**](#_Toc515252788)

[**srtsaiLinked File 9**](#_Toc515252789)

[**rrsaiLinked File 10**](#_Toc515252790)

[**FCFS.py 11**](#_Toc515252791)

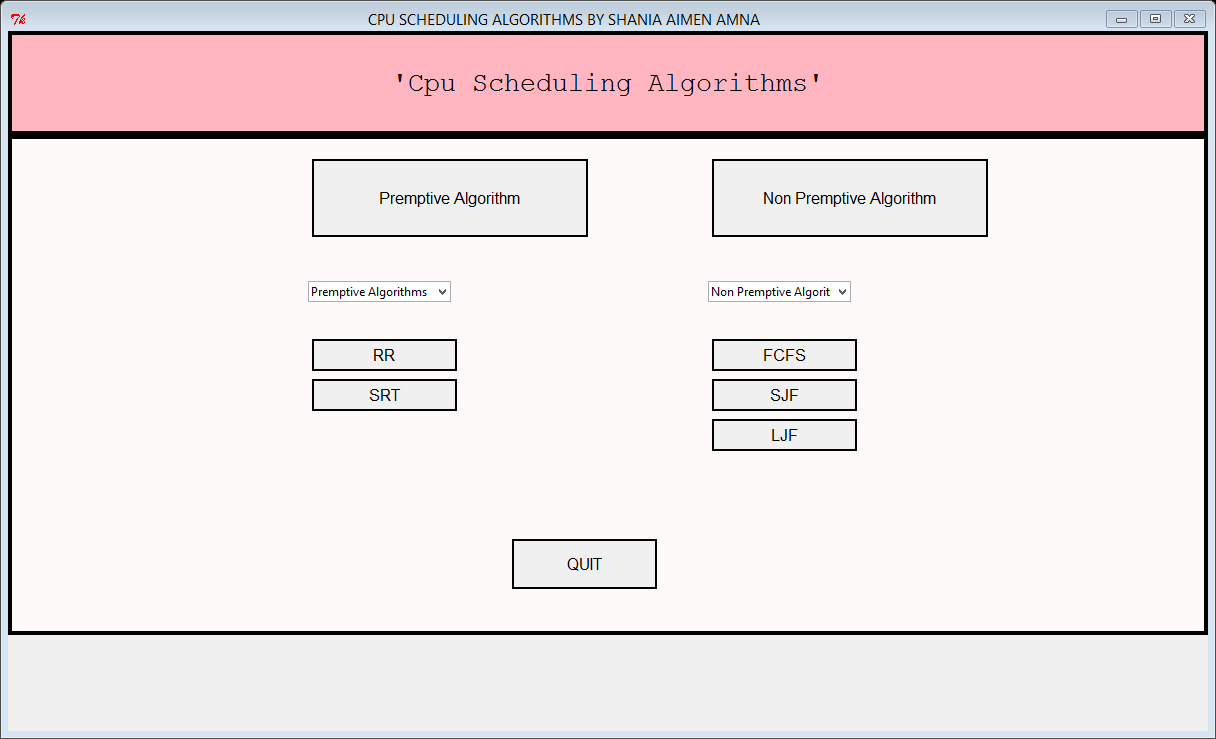
[**SJF.py 12**](#_Toc515252792)

[**LJF.py 14**](#_Toc515252793)

[**SRT.PY 15**](#_Toc515252794)

[**RR.py 22**](#_Toc515252795)

**MAIN GUI**

****

import Tkinter

from Tkinter import \*

import ttk

import os

import sys

'''from FCFS import fcfs

import subprocess'''

win = Tk()

win.geometry("1200x700")

win.title("CPU SCHEDULING ALGORITHMS BY SHANIA KHAN ")

def FCFSwalaFunction():

win.destroy()

os.system('FCFSsaiLinked.py')

def RRwalaFunction():

win.destroy()

os.system('rrsaiLinked.py')

def SRTwalaFunction():

win.destroy()

os.system('srtsaiLinked.py')

def SJFwalaFunction():

win.destroy()

os.system('SJFsaiLinked.py')

def LJFwalaFunction():

win.destroy()

os.system('LJFsaiLinked.py')

topFrame = Frame(win,height=100,bd=4,relief=SOLID)

topFrame.config(bg="light pink")

topFrame.pack(fill=X)

label1= Label (topFrame, text="'Cpu Scheduling Algorithms'", bg="light pink",height=3)

label1.config(font=("Courier",20))

label1.pack()

bottomFrame = Frame(win,height=500,bg="snow",bd=4,relief=SOLID)

#bottomFrame.config(bg="snow")

bottomFrame.pack(fill=X)

label2 =Label (bottomFrame,text="Premptive Algorithm",width=30,height=4,font="bold",relief=SOLID)

label2.place(x=300,y=20)

#label2.pack(side='top')

valuecombo = ('RR','SRT')

varcombo = Tkinter.StringVar()

varcombo.set('Premptive Algorithms')

combo = ttk.Combobox(values=valuecombo,textvariable=varcombo)

combo.place(x=300,y=250)

button2 = Button (bottomFrame,text="RR",width=15,height=1,font="bold",relief=SOLID,command=RRwalaFunction)

button2.place(x=300,y=200)

button3 = Button (bottomFrame,text="SRT",width=15,height=1,font="bold",relief=SOLID,command=SRTwalaFunction)

button3.place(x=300,y=240)

label3=Label (bottomFrame,text="Non Premptive Algorithm",width=30,height=4,font="bold",relief=SOLID)

label3.place(x=700,y=20)

valuecombo2 = ('FCFS','SJF','LJF')

varcombo2 = Tkinter.StringVar()

varcombo2.set('Non Premptive Algorithms')

combo2 = ttk.Combobox(values=valuecombo2,textvariable=varcombo2)

combo2.place(x=700,y=250)

button4 = Button (bottomFrame,text="FCFS",width=15,height=1,font="bold",relief=SOLID,command=FCFSwalaFunction)

button4.place(x=700,y=200)

button5 = Button (bottomFrame,text="SJF",width=15,height=1,font="bold",relief=SOLID,command=SJFwalaFunction)

button5.place(x=700,y=240)

button6 = Button (bottomFrame,text="LJF",width=15,height=1,font="bold",relief=SOLID,command=LJFwalaFunction)

button6.place(x=700,y=280)

button1 = Button (bottomFrame,text="QUIT",width=15,height=2,font="bold",relief=SOLID,command=win.destroy)

button1.place(x=500,y=400)

bottomFrame.pack()

win.mainloop()

# FCFSsaiLinked File

import Tkinter

from Tkinter import \*

import os

import sys

win = Tk()

def FCFSswalaFunction():

win.destroy()

os.system('FCFS.py')

button2 = Button (win,text="Lets Start",width=15,height=1,font="bold",relief=SOLID,command=FCFSswalaFunction)

#button2.place(x=300,y=200)

button2.pack()

win.mainloop()

import Tkinter

# SJFsaiLinked File

from Tkinter import \*

import ttk

import os

import sys

win = Tk()

def SJFwalaFunction():

win.destroy()

os.system('SJF.py')

button2 = Button (win,text="Lets Start",width=15,height=1,font="bold",relief=SOLID,command=SJFwalaFunction)

#button2.place(x=300,y=200)

button2.pack()

win.mainloop()

# LJFsaiLinked File

import Tkinter

from Tkinter import \*

import ttk

import os

import sys

win = Tk()

def LJFwalaFunction():

win.destroy()

os.system('LJF.py')

button2 = Button (win,text="Lets Start",width=15,height=1,font="bold",relief=SOLID,command=LJFwalaFunction)

#button2.place(x=300,y=200)

button2.pack()

win.mainloop()

# SJFsaiLinked File

import Tkinter

from Tkinter import \*

import ttk

import os

import sys

win = Tk()

def SJFwalaFunction():

win.destroy()

os.system('SJF.py')

button2 = Button (win,text="Lets Start",width=15,height=1,font="bold",relief=SOLID,command=SJFwalaFunction)

#button2.place(x=300,y=200)

button2.pack()

win.mainloop()

# srtsaiLinked File

import Tkinter

from Tkinter import \*

import ttk

import os

import sys

win = Tk()

def SRTwalaFunction():

win.destroy()

os.system('srt.py')

button2 = Button (win,text="Lets Start",width=15,height=1,font="bold",relief=SOLID,command=SRTwalaFunction)

#button2.place(x=300,y=200)

button2.pack()

win.mainloop()

# rrsaiLinked File

import Tkinter

from Tkinter import \*

import ttk

import os

import sys

win = Tk()

def RRwalaFunction():

win.destroy()

os.system('rr.py')

button2 = Button (win,text="Lets Start",width=15,height=1,font="bold",relief=SOLID,command=RRwalaFunction)

#button2.place(x=300,y=200)

button2.pack()

win.mainloop()

# FCFS.py

import os

import sys

from Tkinter import \*

TAT=0

process=[]

wait=[]

exe=[]

num=0

num=int(input("Enter number of Processes:"))

for i in range(0,num):

process.append("Process "+repr((i+1)))

exe.append(input("Enter Execution Time: "+repr(process[i])))

print("PROCESS ",process)

print("EXEC-TIME:",exe)

for i in range(0,num):

wait.append(TAT)

TAT=TAT+int(exe[i])

process="TAT :"+repr(TAT)+" "+"Avg WT : "+str(sum(wait)/len(wait))

win = Tk()

win.title("OUTPUT WINDOW")

m2=Message(win,text=process,width=1000,font=('trebuchet MS',15),bg='lightblue').grid(row=15,column=1)

win.mainloop()

# SJF.py

import os

import sys

from Tkinter import \*

process=[]

exe=[]

TAT=0

wait=[]

def sort():

for i in range(0,num):

for j in range(0,num-1):

if(exe[j]>exe[j+1]):

#execution

temp=exe[j]

exe[j]=exe[j+1]

exe[j+1]=temp

#process

temp=process[j]

process[j]=process[j+1]

process[j+1]=temp

print(exe)

print(process)

num=int(input("Enter number of Processes: "))

for i in range(0,num):

process.append(str("Process"+str(i+1)))

exe.append(int(input("Enter EXE-TIME = "+str(process[i])+" :")))

sort()

for i in range(0,num):

wait.append(TAT)

TAT=TAT+exe[i]

process="TAT :"+str(TAT)+" Ave Waiting Time :"+str(sum(wait)/len(wait))

win = Tk()

win.title("OUTPUT WINDOW")

m2=Message(win,text=process,width=1000,font=('trebuchet MS',15),bg='lightblue').grid(row=15,column=1)

win.mainloop()

os.system("pause")

# LJF.py

import os

import sys

from Tkinter import \*

process=[]

exe=[]

TAT=0

wait=[]

def sort():

for i in range(0,num):

for j in range(0,num-1):

if(exe[j]<exe[j+1]):

#execution

temp=exe[j]

exe[j]=exe[j+1]

exe[j+1]=temp

#process

temp=process[j]

process[j]=process[j+1]

process[j+1]=temp

print(exe)

print(process)

num=int(input("Enter number of Processes: "))

for i in range(0,num):

process.append(str("Process"+str(i+1)))

exe.append(int(input("Enter EXE-TIME = "+str(process[i])+" :")))

sort()

for i in range(0,num):

wait.append(TAT)

TAT=TAT+exe[i]

process="Turnaround time :"+str(TAT)+" AWT :"+str(sum(wait)/len(wait))

win = Tk()

win.title("OUTPUT WINDOW")

m2=Message(win,text=process,width=1000,font=('trebuchet MS',15),bg='lightblue').grid(row=15,column=1)

win.mainloop()

os.system("pause")

# SRT.PY

import os

import sys

from Tkinter import \*

tempP=[]#queue proceessess

tempExe=[]#exe time

tempTime=[0]#total time

tempL=[]#hold order of processes received throghout the program

p2=[]

e2=[]

a2=[]

exe=[]

arr=[]

p=[]

wt=[]

def calTime():

for process in p2:

p.append(process)

#start=0

end=-1

waitingTime=0

for j in range(0,len(tempL)):

if(end ==-1 and process==tempL[j]):

waitingTime=tempTime[j]-a2[p2.index(process)]

end=tempTime[j+1]

elif(process==tempL[j]):

waitingTime+=(tempTime[j]-end);

end=tempTime[j+1]

wt.append(waitingTime)

print()

for i in range(0,len(p2)):

print("Waiting time for "+p2[i]+"="+str(wt[i])+

" TurnAround time for "+p2[i]+"="+str(wt[i]+e2[i]))

if(len(wt)>0):

print("\nAverage waiting time="+str(sum(wt)/len(wt)))

def printResult():

pri=""

for i in range(0,len(tempL)):

pri=pri+"("+str(tempTime[i])+") "+tempL[i]+" "

pri+="("+str(tempTime[len(tempTime)-1])+")"

print(pri)

def checkArrival(count):

if(len(arr)>0):

for i in range(0,len(arr)):

if(arr[0]<=count):

tempP.append(p.pop(0))

tempExe.append(exe.pop(0))

arr.pop(0)

else:

break

def exeSRT():

count=0

finished=0

while(len(tempP)==0):

checkArrival(count)

if(len(tempP)==0):

count+=1

while(len(p)>0 or len(tempP)>0):

a=len(tempP)

t=0

m=tempExe.index(min(tempExe))

#print("t="+str(t)+" a="+str(a)+"m="+str(m))

while (1):

if(len(p)>0):

#print("if len(p)>0. len(p)="+str(len(p))+" p="+tempP[m]+" exe="+

# str(tempExe[m]))

t+=1

tempExe[m]=tempExe[m]-1

count+=1

checkArrival(count)

if(len(tempP)>a):

#print("if(len(tempP)>a) t="+str(t))

tempL.append(tempP[m])

tempTime.append(tempTime[len(tempTime)-1]+t)

#checkArrival(count)

break

elif(tempExe[m]==0):

#print("elif(tempExe[m]==0)")

tempL.append(tempP[m])

tempTime.append(tempTime[len(tempTime)-1]+t)

tempP.pop(m)

tempExe.pop(m)

#this while is not necessecary if arrival time don't

#have gaps

while(len(tempP)==0):

checkArrival(count)

count+=1

#checkArrival(count)

break

else:

while(len(tempP)>0):

m=tempExe.index(min(tempExe))

tempL.append(tempP[m])

tempTime.append(tempTime[len(tempTime)-1]+tempExe[m])

tempP.pop(m)

tempExe.pop(m)

break

printResult()

calTime()

def sortByArr():

for i in range(0,num):

for j in range(0,num-1):

if(arr[j]>arr[j+1]):

#for arr

temp=arr[j]

arr[j]=arr[j+1]

arr[j+1]=temp

#for pname

temp=p[j]

p[j]=p[j+1]

p[j+1]=temp

#for exe

temp=exe[j]

exe[j]=exe[j+1]

exe[j+1]=temp

print()

print("Processes: "+str(p))

print("Exe Time: "+str(exe))

print("Arrival Time: "+str(arr))

print()

tempExe=exe

#print("\nLength of tempExe="+str(len(tempExe)))

def appendData():

for i in range(0,len(p)):

p2.append(p[i])

a2.append(arr[i])

e2.append(exe[i])

num=int(input("Enter number of processess: "))

for i in range(0,num):

p.append(str("P"+str(i+1)))

arr.append(int(input("Enter arrival time of "+str(p[i])+" :")))

exe.append(int(input("Enter exe time of "+str(p[i])+" :")))

#tS=int(input("Enter timeslice: "))

print()

appendData()

sortByArr()

exeSRT()

os.system("pause")

# RR.py

import os

import sys

from Tkinter import \*

tempP=[]#queue proceessess

tempExe=[]#exe time

tempTime=[0]#total time

tempL=[]#hold order of processes received throghout the program

p2=[]

e2=[]

a2=[]

exe=[]

arr=[]

p=[]

tS=0 #time slice

#TAT=0

wt=[]

def calTime():

for process in p2:

p.append(process)

#start=0

end=-1

waitingTime=0

for j in range(0,len(tempL)):

if(end ==-1 and process==tempL[j]):

waitingTime=tempTime[j]-a2[p2.index(process)]

end=tempTime[j+1]

elif(process==tempL[j]):

waitingTime+=(tempTime[j]-end);

end=tempTime[j+1]

wt.append(waitingTime)

print()

for i in range(0,len(p2)):

print("Waiting time for "+p2[i]+"="+str(wt[i])+

" TurnAround time for "+p2[i]+"="+str(wt[i]+e2[i]))

if(len(wt)>0):

print("\nAverage waiting time="+str(sum(wt)/len(wt)))

def printResult():

pri=""

for i in range(0,len(tempL)):

pri=pri+"("+str(tempTime[i])+") "+tempL[i]+" "

pri+="("+str(tempTime[len(tempTime)-1])+")"

print(pri)

def checkArrival(count):

if(len(arr)>0):

for i in range(0,len(arr)):

if(arr[0]<=count):

tempP.append(p.pop(0))

tempExe.append(exe.pop(0))

arr.pop(0)

else:

break

def exeRR():

count=0

total=len(p)

while(1):

if(len(arr)>0 and arr[0]==count):

checkArrival(count)

if(len(tempP)>0):

if(tempExe[0]>tS):

tempExe[0]=tempExe[0]-tS;

tempTime.append(tempTime[len(tempTime)-1]+tS);

tempL.append(tempP[0])

count+=tS

checkArrival(count)

tempP.append(tempP.pop(0))

tempExe.append(tempExe.pop(0))

elif(tempExe[0]>0):

tempTime.append(tempTime[len(tempTime)-1]+tempExe[0]);

tempL.append(tempP[0])

count+=tempExe[0]

tempExe.pop(0)

tempP.pop(0)

else:

tempExe.pop(0)

tempP.pop(0)

if(len(p)==0 and len(tempP)==0):

printResult()

calTime()

break

#printResult()

#calTime()

def sortByArr():

for i in range(0,num):

for j in range(0,num-1):

if(arr[j]>arr[j+1]):

#for arr

temp=arr[j]

arr[j]=arr[j+1]

arr[j+1]=temp

#for pname

temp=p[j]

p[j]=p[j+1]

p[j+1]=temp

#for exe

temp=exe[j]

exe[j]=exe[j+1]

exe[j+1]=temp

print()

print("Processes: "+str(p))

print("Exe Time: "+str(exe))

print("Arrival Time: "+str(arr))

print()

tempExe=exe

def appendData():

for i in range(0,len(p)):

p2.append(p[i])

a2.append(arr[i])

e2.append(exe[i])

num=int(input("Enter number of processess: "))

for i in range(0,num):

p.append(str("P"+str(i+1)))

arr.append(int(input("Enter arrival time of "+str(p[i])+" :")))

exe.append(int(input("Enter exe time of "+str(p[i])+" :")))

tS=int(input("Enter timeslice: "))

print()

appendData()

sortByArr()

exeRR()