

Centro Universitario de Ciencias Exactas e Ingenierías

Departamento de Ciencias Computacionales

Sistemas Operativos



Profesora: Becerra Velázquez Violeta del Rocío

Alumno: Monjaraz Briseño Luis Fernando

Código: 218520958

Carrera: Ingeniería en Computación

Sección: D04

Actividad 16 (Programa 8 Procesos suspendidos)

Fecha: 03/12/2023

Índice

Índice	2
Tabla de imágenes	3
Datos personales	4
Datos de la materia	4
Número de actividad	4
Objetivo de la actividad	4
Notas acerca del lenguaje	4
Conclusión	21
Lomeli Jaime:	21
Monjaraz Luis:	21

Tabla de imágenes

Ilustración 1 setupUi	6
Ilustración 2 retranslateUi	7
Ilustración 3 init	8
Ilustración 4 start_stop_func	8
Ilustración 5 pedirProcesos	9
Ilustración 6 terminadoProceso	10
Ilustración 7 inicio	10
Ilustración 8 paginacion	11
Ilustración 9 suspendido	12
Ilustración 10 regresar	13
Ilustración 11 ejecucion	14
Ilustración 12 init 2	14
Ilustración 13 keyPressEvent	15
Ilustración 14 setupUi 2	16
Ilustración 15 retranslateUi 2	17
Ilustración 16 Bloqueamos las teclas de tabulador y espacio para evitar errores.	17
Ilustración 17 Impresión de suspendidos en BCP.	17
Ilustración 18 Teclas	18
Ilustración 19 Ejemplo	19
Ilustración 20 Ejemplo 2	19
Ilustración 21 BCP	20
Ilustración 22 Txt	20

Datos personales

Nombres: Lomeli Navarro Jaime Adalberto y Monjaraz Briseño Luis Fernando.

Códigos: 221978094 y 218520958.

Correos: jaime.lomeli7809@alumnos.udg.mx y luis.monjaraz5209@alumnos.udg.mx

Datos de la materia

Materia: Sistemas Operativos

Sección: D04

Horario: Martes, Jueves, Sábado. 11:00 a 12:55

NRC: 204880

Clave: IL366

Número de actividad

Programa 8 Procesos suspendidos

Objetivo de la actividad

El objetivo de esta actividad es mejorar el algoritmo de paginación simple pero ahora con “suspendidos”, esto consta del mismo funcionamiento del Round Robin, que se menciona en el programa de Paginación simple, solo que ahora se agregan dos nuevas teclas, siendo S y R, S sirviendo para suspender los procesos que estén interrumpidos siendo enviados a disco, y R para regresar a la memoria. Esto para simular el cuando un proceso es mandado a alguna parte de nuestro disco, en este caso será un TXT, en el que almacenaremos la ID.

En resumen, esta actividad es sumamente útil para comprender la teoría de la paginación simple con paginación y el concepto del disco.

Notas acerca del lenguaje

Lenguaje usado: Python.

Motivo: Principalmente escogimos este lenguaje por que decidimos trabajar en equipo para esta actividad, y al ser el lenguaje que mejor sabemos usar los dos se llegó a la conclusión de utilizarlo. También es por que es una continuación directa de paginación simple, por lo que solo tuvimos que modificar pequeñas cosas. Además, Python cuenta con muchas herramientas que facilitan mucho la programación de la simulación de los procesos, además, aprovechando que en Análisis de algoritmos hemos estado trabajando con la librería “PyQT” para interfaces graficas decidimos emplearla para nuestra actividad, cosa que consideramos que fue todo un éxito.

Estructuras: Utilizamos “clases” para la resolución del problema, una llamada “Ui_widget” y “MainWindow”. En estas clases la crearon varias funciones. La clase Ui_widget es la que se encarga principalmente del funcionamiento del programa, esto a nivel interno y

visual, mientras que la clase “MainWindows” se encarga más que nada de las pulsaciones de las teclas y otros detalles. Además de eso se utilizó otro archivo Python solamente para el BCP, en este si utilizo la clase “Ui_Form”.

El código es de $1603 + 96 = 1699$ líneas de código.

Funciones:

- setupUi: Esta es principalmente para el diseño de la interfaz con PyQt.
- retranslateUi: Esta es principalmente para el diseño, sin embargo, este se ira actualizando conforme avance el programa.
- __init__: Esta función es para inicializar algunas variables.
- start_stop_func: Inicializamos tiempos de bloqueo.
- pedirProcesos: Esta función es para solicitar la cantidad de procesos y asignarle sus respectivos datos de forma aleatoria.
- terminadoProceso: Esta función es la que se encarga de imprimir y manejar los procesos terminados.
- Inicio: Inicializa y controla contadores.
- Paginación: Esta función controla todo el funcionamiento de la paginación simple.
- Suspendido: Esta función se encarga de la suspensión, sin embargo, esto es para la próxima tarea, debido a que necesita correcciones, las cuales nos mencionó en clase.
- Regresar: Esta función se encarga de sacar a los procesos de suspendidos.
- Ejecución: Esta función se encarga de los procesos en ejecución.
- __init__: Inicializa variables de la segunda clase.
- keyPressEvent: Esta función se encarga de “recibir” las teclas que son pulsadas.
- setupUi: Esta función es para el diseño de la interfaz del BCP.
- retranslateUi: Esta función ajusta y imprime los valores del BCP.

Para el Quantum se creó una variable global llamada Quantum y a esta se le creo una copia, esto

Funciones:

```

25 def setupUi(self, widget):
26     widget.setObjectName("widget")
27     widget.resize(1921, 1019)
28     widget.setStyleSheet("QWidget#widget{\n"
29         "background-color: rgb(255, 239, 213);}")
30     self.label = QtWidgets.QLabel(widget)
31     self.label.setGeometry(QtCore.QRect(660, 10, 291, 40))
32     self.label.setStyleSheet("font: 20pt \"Algerian\";\n"
33         "\"background-color: rgb(255, 255, 255);")
34     self.label.setObjectName("Label")
35     self.label_2 = QtWidgets.QLabel(widget)
36     self.label_2.setGeometry(QtCore.QRect(10, 30, 221, 31))
37     self.label_2.setStyleSheet("font: 14pt \"Times New Roman\";")
38     self.label_2.setObjectName("Label_2")
39     self.proces = QtWidgets.QLineEdit(widget)
40     self.proces.setGeometry(QtCore.QRect(240, 30, 71, 31))
41     self.proces.setStyleSheet("font: 14pt \"Times New Roman\";")
42     self.proces.setText("")
43     self.proces.setObjectName("proces")
44     self.iniciar = QtWidgets.QPushButton(widget)
45     self.iniciar.setGeometry(QtCore.QRect(320, 70, 141, 31))
46     self.iniciar.setStyleSheet("font: 14pt \"Times New Roman\";")
47     self.iniciar.setObjectName("iniciar")
48
49     self.iniciar.clicked.connect(self.pedirProcesos)
50     self.iniciar.clicked.connect(self.inicio)
51     self.iniciar.clicked.connect(self.start_stop_func)
52
53     self.label_3 = QtWidgets.QLabel(widget)
54     self.label_3.setGeometry(QtCore.QRect(90, 160, 177, 31))
55     self.label_3.setStyleSheet("font: 14pt \"Times New Roman\";")
56     self.label_3.setObjectName("Label_3")
57     self.Nuevos = QtWidgets.QLabel(widget)
58     self.Nuevos.setGeometry(QtCore.QRect(280, 160, 131, 31))
59     self.Nuevos.setStyleSheet("font: 14pt \"Times New Roman\";\n"
60         "\"background-color: rgb(255, 255, 255);")
61     self.Nuevos.setText("")
62     self.Nuevos.setObjectName("Nuevos")
63     self.label_4 = QtWidgets.QLabel(widget)
64     self.label_4.setGeometry(QtCore.QRect(160, 460, 137, 31))
65     self.label_4.setStyleSheet("font: 14pt \"Times New Roman\";\n"
66         "\"background-color: rgb(255, 255, 255);")
67     self.label_4.setObjectName("Label_4")
68     self.tablaListos = QtWidgets.QTableWidget(widget)
69     self.tablaListos.setGeometry(QtCore.QRect(20, 500, 421, 231))
70     self.tablaListos.setStyleSheet("font: 10pt \"Times New Roman\";")
71     self.tablaListos.setObjectName("tablaListos")
72     self.tablaListos.setColumnCount(3)
73     self.tablaListos.setRowCount(0)
74     item = QtWidgets.QTableWidgetItem()
75     self.tablaListos.setHorizontalHeaderItem(0, item)
76     item = QtWidgets.QTableWidgetItem()
77     self.tablaListos.setHorizontalHeaderItem(1, item)
78     item = QtWidgets.QTableWidgetItem()

```

Ilustración 1 setupUi

```

520 def retranslateUi(self, widget):
521     _translate = QtCore.QCoreApplication.translate
522     widget.setWindowTitle(_translate("widget", "Form"))
523     self.label.setText(_translate("widget", "PS Suspendidos"))
524     self.label_2.setText(_translate("widget", "Número de procesos:"))
525     self.iniciar.setText(_translate("widget", "Iniciar"))
526     self.label_3.setText(_translate("widget", "Procesos nuevos:"))
527     self.label_4.setText(_translate("widget", "Cola de listos"))
528     item = self.tablaListos.horizontalHeaderItem(0)
529     item.setText(_translate("widget", "ID"))
530     item = self.tablaListos.horizontalHeaderItem(1)
531     item.setText(_translate("widget", "TME"))
532     item = self.tablaListos.horizontalHeaderItem(2)
533     item.setText(_translate("widget", "TT"))
534     self.label_5.setText(_translate("widget", "Proceso en ejecución"))
535     self.label_6.setText(_translate("widget", "Número de programa (ID):"))
536     self.label_7.setText(_translate("widget", "Operación a realizar:"))
537     self.label_8.setText(_translate("widget", "Tiempo maximo estimado:"))
538     self.label_9.setText(_translate("widget", "Tiempo transcurrido:"))
539     self.label_10.setText(_translate("widget", "Tiempo restante:"))
540     self.label_11.setText(_translate("widget", "Procesos Bloqueados"))
541     item = self.bloqueados.horizontalHeaderItem(0)
542     item.setText(_translate("widget", "ID"))
543     item = self.bloqueados.horizontalHeaderItem(1)
544     item.setText(_translate("widget", "TT"))
545     self.label_12.setText(_translate("widget", "Procesos terminados"))
546     item = self.Terminados.horizontalHeaderItem(0)
547     item.setText(_translate("widget", "ID"))
548     item = self.Terminados.horizontalHeaderItem(1)
549     item.setText(_translate("widget", "Operación"))
550     item = self.Terminados.horizontalHeaderItem(2)
551     item.setText(_translate("widget", "Resultado"))
552     self.label_13.setText(_translate("widget", "Reloj:"))
553     self.label_14.setText(_translate("widget", "Quantum:"))
554     self.label_15.setText(_translate("widget", "Quantum:"))
555     self.label_16.setText(_translate("widget", "Valor del Quantum:"))
556     self.label_17.setText(_translate("widget", "Proceso a entrar"))
557     self.label_18.setText(_translate("widget", "ID:"))
558     self.label_19.setText(_translate("widget", "Tamaño:"))
559     item = self.tableWidget.verticalHeaderItem(0)
560     item.setText(_translate("widget", "0"))
561     item = self.tableWidget.verticalHeaderItem(1)
562     item.setText(_translate("widget", "1"))
563     item = self.tableWidget.verticalHeaderItem(2)
564     item.setText(_translate("widget", "2"))
565     item = self.tableWidget.verticalHeaderItem(3)
566     item.setText(_translate("widget", "3"))
567     item = self.tableWidget.verticalHeaderItem(4)
568     item.setText(_translate("widget", "4"))
569     item = self.tableWidget.verticalHeaderItem(5)
570     item.setText(_translate("widget", "5"))
571     item = self.tableWidget.verticalHeaderItem(6)

```

Ilustración 2 retranslateUi

Procesos suspendidos

```
667
668     def __init__(self):
669         self.step = 0
670
671         self.timer = QTimer(self)
672         self.timer.timeout.connect(self.ejecucion)
673
674
```

Ilustración 3 init

```
674
675     def start_stop_func(self):
676         if not self.timer.isActive():
677             self.timer.start(800)
678         else:
679             self.timer.stop()
680
```

Ilustración 4 start_stop_func


```

716
717 def pedirProcesos(self):
718     try:
719         proses = int(self.proces.text())
720         self.quantum = int(self.quant.text())
721         _translate = QtCore.QCoreApplication.translate
722         self.proces.setText(_translate("Dialog", ""))
723         self.quant.setText(_translate("Dialog", ""))
724
725     except ValueError:
726         print("El valor ingresado no es un número, vuelva a ingresar")
727
728     contador = 0
729
730     tme = 0
731     oper = 0
732
733     _translate = QtCore.QCoreApplication.translate
734     self.valorQuant.setText(_translate("Dialog", str(self.quantum)))
735
736     while contador < proses:
737         lista = []
738
739         lista.append(self.ids)
740
741         tme = random.randint(6, 16)
742         lista.append(tme)
743
744         lista.append(0)
745
746         oper = random.randint(1, 6)
747         if oper == 1:
748             num1 = random.randint(0, 1000)
749             num2 = random.randint(0, 1000)
750             operacion = str(num1) + "+" + str(num2)
751             resultado = num1 + num2
752         elif oper == 2:
753             num1 = random.randint(0, 1000)
754             num2 = random.randint(0, 1000)
755             operacion = str(num1) + "-" + str(num2)
756             resultado = num1 - num2
757         elif oper == 3:
758             num1 = random.randint(-1000, 1000)

```

Ilustración 5 pedirProcesos

```
839
840 def terminadoProceso(self):
841     if self.lisd == [] and len(self.listaAux) > 0:
842         self.quantum2 = 1
843         self.lisd = self.listaAux.pop(0)
844     try:
845         if self.lisd[7] == 0:
846             self.lisd[7] = 1
847             self.lisd[6] = self.contadorGlobal
848     except IndexError:
849         pass
850
851     try:
852         if self.lisd[2] != 0:
853             self.cont = self.lisd[2] + 1
854         else:
855             self.cont = 1
856     except IndexError:
857         pass
858     self.terminador = 1
859
860     if len(self.listaLotes) == self.contPros:
861         self.bcp=QtWidgets.QDialog()
862         self.ui=Ui_Form()
863         self.ui.setupUi(self.bcp)
864         self.bcp.show()
865         cuenta = 0
866
867         self.ui.BCP.setRowCount(len(self.terminados))
868         for i in self.terminados:
869             self.ui.BCP.setItem(cuenta, 0, QtWidgets.QTableWidgetItem(str(i[0])))
870             self.ui.BCP.setItem(cuenta, 1, QtWidgets.QTableWidgetItem(str(i[1])))
871             self.ui.BCP.setItem(cuenta, 2, QtWidgets.QTableWidgetItem(str(i[2])))
872             self.ui.BCP.setItem(cuenta, 3, QtWidgets.QTableWidgetItem(str(i[3])))
873             self.ui.BCP.setItem(cuenta, 4, QtWidgets.QTableWidgetItem(str(i[4])))
874             self.ui.BCP.setItem(cuenta, 5, QtWidgets.QTableWidgetItem(str(i[5])))
875             self.ui.BCP.setItem(cuenta, 6, QtWidgets.QTableWidgetItem(str(i[6])))
876             self.ui.BCP.setItem(cuenta, 7, QtWidgets.QTableWidgetItem(str(i[7])))
877             cuenta = cuenta + 1
```

Ilustración 6 terminadoProceso

```
883
884 def inicio(self):
885     try:
886         if self.lisd[2] != 0:
887             self.cont = self.lisd[2]
888         else:
889             self.cont = 1
890     except IndexError:
891         pass
892     self.terminador = 1
893
```

Ilustración 7 inicio

```

893
894     def paginacion(self):
895         try:
896             for x in range(len(self.memor)):
897                 if self.memor[x] == self.lisd[0]:
898                     lisAyuda = []
899                     lisAyuda.append(str(self.memor1[x]) + "/5")
900                     lisAyuda.append(self.memor[x])
901                     lisAyuda.append("Ejecucion")
902                     self.memor2[x] = lisAyuda
903         except IndexError:
904             pass
905         try:
906             for x in range(len(self.memor)):
907                 for i in self.listaAux:
908                     if self.memor[x] == i[0]:
909                         lisAyuda = []
910                         lisAyuda.append(str(self.memor1[x]) + "/5")
911                         lisAyuda.append(self.memor[x])
912                         lisAyuda.append("Listo")
913                         self.memor2[x] = lisAyuda
914         except IndexError:
915             pass
916         try:
917             for x in range(len(self.memor)):
918                 for i in self.prosBloq:
919                     if self.memor[x] == i[0]:
920                         lisAyuda = []
921                         lisAyuda.append(str(self.memor1[x]) + "/5")
922                         lisAyuda.append(self.memor[x])

```

Ilustración 8 paginacion

```
957
958     def suspendido(self):
959         try:
960             listaP = self.prosBloq.pop(0)
961             self.suma2 = self.suma2 - listaP[14]
962             self.suspen.append(listaP)
963             for x in range(len(self.memor)):
964                 if self.memor[x] == listaP[0]:
965                     self.memor[x] = 0
966                     self.memor1[x] = 0
967         try:
968             while True:
969                 nuevalis = self.otraLista.pop(0)
970                 print(nuevalis)
971                 if nuevalis[14] <= 40 - self.suma2:
972                     memoria3 = nuevalis[14]
973                     memoria4 = nuevalis[13]
974                     memoria5 = int(memoria4/5)
975                     memoria4 = memoria4%5
976                     for x in range(len(self.memor)):
977                         if memoria3 == 0:
978                             break
979                         if self.memor[x] == 0:
980                             self.memor[x] = nuevalis[0]
981                             if memoria5 > 0:
982                                 self.memor1[x] = 5
983                                 memoria5 -= 1
984                             else:
985                                 self.memor1[x] = memoria4
986                                 memoria3 -= 1
987
988                     self.suma2 = self.suma2 + nuevalis[14]
989                     nuevalis[5] = self.contadorGlobal
990                     self.listaAux.append(nuevalis)
991                 else:
992                     self.otraLista = [nuevalis] + self.otraLista
993                     break
994         except IndexError:
995             pass
996     except IndexError:
997         pass
998
```

Ilustración 9 suspendido

```

1008
1009 def regresar(self):
1010     try:
1011         lista = self.suspen.pop(0)
1012         print(lista)
1013
1014         tamano2 = int(lista[13]/5)
1015         tamano3 = tamano2
1016         tamano4 = lista[13]%5
1017         print(tamano2)
1018         self.suma = self.suma2
1019         self.suma = self.suma + tamano2
1020         if tamano4 != 0:
1021             self.suma = self.suma + 1
1022             tamano2 += 1
1023         if self.suma <= 40:
1024             self.suma2 = self.suma
1025             self.prosBloq.append(lista)
1026             for x in range(len(self.memor)):
1027                 if tamano2 == 0:
1028                     break
1029                 if self.memor[x] == 0:
1030                     self.memor[x] = lista[0]
1031                     if tamano3 > 0:
1032                         self.memor1[x] = 5
1033                         tamano3 -= 1
1034                     else:
1035                         self.memor1[x] = tamano4
1036                         tamano2 -= 1
1037             else:
1038                 self.suspen = [lista] + self.suspen
1039     except IndexError:
1040         pass
1041
1042     archivo = open("Procesos en Disco.txt", "w", encoding='utf8')
1043     try:
1044         for c in self.suspen:
1045             archivo.write("ID: ")
1046             archivo.write(str(c[0]))
1047             archivo.write("\n")
1048         archivo.close()
1049     except:
1050         pass
1051

```

Ilustración 10 regresar

```

1052     def ejecucion(self):
1053         _translate = QtCore.QCoreApplication.translate
1054         self.step += 1
1055         if self.bandera == 3:
1056             self.bandera = 0
1057             self.start_stop_func()
1058         elif self.bandera == 6:
1059             self.bandera = 0
1060             self.suspendido()
1061         elif self.bandera == 7:
1062             self.bandera = 0
1063             self.regresar()
1064
1065         try:
1066             listoentrar = self.otraLista[0]
1067             self.Nuevos_2.setText(_translate("widget", str(listoentrar[0])))
1068             self.Nuevos_3.setText(_translate("widget", str(listoentrar[13])))
1069             self.Nuevos_4.setText(_translate("widget", str(listoentrar[14])))
1070         except IndexError:
1071             self.Nuevos_2.setText(_translate("widget", str(0)))
1072             self.Nuevos_3.setText(_translate("widget", str(0)))
1073             self.Nuevos_4.setText(_translate("widget", str(0)))
1074
1075         try:
1076             listoentrar = self.suspen[0]
1077             medida = len(self.suspen)
1078             self.disco1.setText(_translate("widget", str(medida)))
1079             self.disco2.setText(_translate("widget", str(listoentrar[0])))
1080             self.disco3.setText(_translate("widget", str(listoentrar[13])))
1081             self.disco4.setText(_translate("widget", str(listoentrar[14])))
1082         except IndexError:
1083             self.disco1.setText(_translate("widget", str(0)))
1084             self.disco2.setText(_translate("widget", str(0)))
1085             self.disco3.setText(_translate("widget", str(0)))
1086             self.disco4.setText(_translate("widget", str(0)))
1087
1088         for p in self.prosBloq:
1089             if p[15] == 8:
1090                 p.pop(8)
1091                 self.listaAux.append(p)
1092                 for n in self.prosBloq:
1093                     if n[0] == p[0]:
1094                         self.prosBloq.remove(n)
1095
1096         trabajos = len(self.otraLista)
1097         self.Nuevos.setText(_translate("widget", str(trabajos)))
1098

```

Ilustración 11 ejecucion

```

class MainWindow(QMainWindow, Ui_widget):
    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        self.setupUi(self)

```

Ilustración 12 init 2

```
def keyPressEvent(self, event):  
    if event.key() == Qt.Key_I:  
        self.bandera = 1  
        print("Usted a presionado I")  
    elif event.key() == Qt.Key_E:  
        self.bandera = 2  
        print("Usted a presionado E")  
    elif event.key() == Qt.Key_P:  
        self.bandera = 3  
        print("Usted a presionado P")  
    elif event.key() == Qt.Key_T:  
        self.bandera = 3  
        print("Usted a presionado T")  
    elif event.key() == Qt.Key_N:  
        self.bandera = 4  
        print("Usted a presionado N")  
    elif event.key() == Qt.Key_B:  
        self.bandera = 5  
        print("Usted a presionado B")  
    elif event.key() == Qt.Key_C:  
        self.start_stop_func()  
        print("Usted a presionado C")  
    elif event.key() == Qt.Key_S:  
        self.bandera = 6  
        print("Usted a presionado S")  
    elif event.key() == Qt.Key_R:  
        self.bandera = 7  
        print("Usted a presionado R")
```

Ilustración 13 keyPressEvent

```
15     def setupUi(self, Form):
16         Form.setObjectName("Form")
17         Form.resize(1535, 596)
18         Form.setStyleSheet("background-color: rgb(255, 239, 213);")
19         self.verticalLayout = QtWidgets.QVBoxLayout(Form)
20         self.verticalLayout.setObjectName("verticalLayout")
21         self.label = QtWidgets.QLabel(Form)
22         self.label.setStyleSheet("font: 20pt \"Algerian\";")
23         self.label.setObjectName("label")
24         self.verticalLayout.addWidget(self.label)
25         self.BCP = QtWidgets.QTableWidget(Form)
26         self.BCP.setStyleSheet("font: 10pt \"Times New Roman\";")
27         self.BCP.setObjectName("BCP")
28         self.BCP.setColumnCount(12)
29         self.BCP.setRowCount(0)
30         item = QtWidgets.QTableWidgetItem()
31         self.BCP.setHorizontalHeaderItem(0, item)
32         item = QtWidgets.QTableWidgetItem()
33         self.BCP.setHorizontalHeaderItem(1, item)
34         item = QtWidgets.QTableWidgetItem()
35         self.BCP.setHorizontalHeaderItem(2, item)
36         item = QtWidgets.QTableWidgetItem()
37         self.BCP.setHorizontalHeaderItem(3, item)
38         item = QtWidgets.QTableWidgetItem()
39         self.BCP.setHorizontalHeaderItem(4, item)
40         item = QtWidgets.QTableWidgetItem()
41         self.BCP.setHorizontalHeaderItem(5, item)
42         item = QtWidgets.QTableWidgetItem()
43         self.BCP.setHorizontalHeaderItem(6, item)
44         item = QtWidgets.QTableWidgetItem()
45         self.BCP.setHorizontalHeaderItem(7, item)
46         item = QtWidgets.QTableWidgetItem()
47         self.BCP.setHorizontalHeaderItem(8, item)
48         item = QtWidgets.QTableWidgetItem()
49         self.BCP.setHorizontalHeaderItem(9, item)
50         item = QtWidgets.QTableWidgetItem()
51         self.BCP.setHorizontalHeaderItem(10, item)
52         item = QtWidgets.QTableWidgetItem()
53         self.BCP.setHorizontalHeaderItem(11, item)
54         self.verticalLayout.addWidget(self.BCP)
55
56         self.retranslateUi(Form)
57         QtCore.QMetaObject.connectSlotsByName(Form)
58
```

Ilustración 14 setupUi 2


```

59     def retranslateUi(self, Form):
60         _translate = QtCore.QCoreApplication.translate
61         Form.setWindowTitle(_translate("Form", "Form"))
62         self.label.setText(_translate("Form", "Tabla de procesos"))
63         item = self.BCP.horizontalHeaderItem(0)
64         item.setText(_translate("Form", "ID"))
65         item = self.BCP.horizontalHeaderItem(1)
66         item.setText(_translate("Form", "Operación"))
67         item = self.BCP.horizontalHeaderItem(2)
68         item.setText(_translate("Form", "Resultado"))
69         item = self.BCP.horizontalHeaderItem(3)
70         item.setText(_translate("Form", "Estado"))
71         item = self.BCP.horizontalHeaderItem(4)
72         item.setText(_translate("Form", "TME"))
73         item = self.BCP.horizontalHeaderItem(5)
74         item.setText(_translate("Form", "Llegada"))
75         item = self.BCP.horizontalHeaderItem(6)
76         item.setText(_translate("Form", "Finalización"))
77         item = self.BCP.horizontalHeaderItem(7)
78         item.setText(_translate("Form", "Retorno"))
79         item = self.BCP.horizontalHeaderItem(8)
80         item.setText(_translate("Form", "Respuesta"))
81         item = self.BCP.horizontalHeaderItem(9)
82         item.setText(_translate("Form", "Servicio"))
83         item = self.BCP.horizontalHeaderItem(10)
84         item.setText(_translate("Form", "Espera"))
85         item = self.BCP.horizontalHeaderItem(11)
86         item.setText(_translate("Form", "Restante"))

```

Ilustración 15 retranslateUi 2

Suspendidos (agregados)

```

23     keyboard.block_key('tab')
24     keyboard.block_key('space')
25

```

Ilustración 16 Bloqueamos las teclas de tabulador y espacio para evitar errores.

```

1377     #Suspendidos en bcp
1378     for i in self.suspen:
1379         listo = []
1380         listo.append(i[0])
1381         listo.append(i[3])
1382         listo.append("Null")
1383         listo.append("Suspendido")
1384         listo.append(i[1])
1385         listo.append(i[5])
1386         listo.append("Null")
1387         listo.append("Null")
1388         listo.append(i[6] - listo[5])
1389         listo.append(i[2])
1390         listo.append((self.contadorGlobal - i[5]) - i[2])
1391         listo.append(listo[4] - listo[9])
1392         listaBcp.append(listo)

```

Ilustración 17 Impresión de suspendidos en BCP.

```
1565
1566     def keyPressEvent(self, event):
1567         if event.key() == Qt.Key_I:
1568             self.bandera = 1
1569             print("Usted a presionado I")
1570         elif event.key() == Qt.Key_E:
1571             self.bandera = 2
1572             print("Usted a presionado E")
1573         elif event.key() == Qt.Key_P:
1574             self.bandera = 3
1575             print("Usted a presionado P")
1576         elif event.key() == Qt.Key_T:
1577             self.bandera = 3
1578             print("Usted a presionado T")
1579         elif event.key() == Qt.Key_N:
1580             self.bandera = 4
1581             print("Usted a presionado N")
1582         elif event.key() == Qt.Key_B:
1583             self.bandera = 5
1584             print("Usted a presionado B")
1585         elif event.key() == Qt.Key_C:
1586             self.start_stop_func()
1587             print("Usted a presionado C")
1588         elif event.key() == Qt.Key_S:
1589             self.bandera = 6
1590             print("Usted a presionado S")
1591         elif event.key() == Qt.Key_R:
1592             self.bandera = 7
1593             print("Usted a presionado R")
1594         else:
1595             print("Usted a presionado una tecla invalida")
1596             self.bandera = 0
1597
```

Ilustración 18 Teclas

Como se debería de ver:

Número de procesos:

Quantum:

Valor del Quantum:

Procesos nuevos:

Reloj:

Iniciar

Proceso a entrar

ID: Tamaño: Marcos:

Procesos en Disco

Cantidad en disco

ID: Tamaño: Marcos:

Cola de listos

ID TME TT

Procesos Bloqueados

ID TT

PS SUSPENDIDOS

Proceso en ejecución

Número de programa (ID):

Operación a realizar:

Tiempo maximo estimado:

Tiempo transcurrido:

Tiempo restante:

Quantum:

Procesos terminados

ID Operación Resultado

Marco Tamaño ID Estado

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Marco Tamaño ID Estado

25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

Ilustración 19 Ejemplo

Número de procesos:

Quantum:

Valor del Quantum:

Procesos nuevos:

Reloj: 13

Iniciar

Proceso a entrar

ID: 14 Tamaño: 21 Marcos: 5

Procesos en Disco

Cantidad en disco 2

ID: 1 Tamaño: 16 Marcos: 4

Cola de listos

ID TME TT

Procesos Bloqueados

ID TT

PS SUSPENDIDOS

Proceso en ejecución

Número de programa (ID):

Operación a realizar:

Tiempo maximo estimado:

Tiempo transcurrido:

Tiempo restante:

Quantum:

Procesos terminados

ID Operación Resultado

Marco Tamaño ID Estado

0 5/5 12 Listo

1 5/5 12 Listo

2 5/5 12 Listo

3 2/5 12 Listo

4 5/5 13 Listo

5 5/5 13 Listo

6 5/5 3 Bloqueado

7 5/5 3 Bloqueado

8 5/5 3 Bloqueado

9 2/5 3 Bloqueado

10 5/5 4 Ejecucion

11 5/5 4 Ejecucion

12 5/5 4 Ejecucion

13 5/5 4 Ejecucion

14 3/5 4 Ejecucion

15 5/5 5 Listo

16 5/5 5 Listo

17 5/5 5 Listo

18 5/5 5 Listo

19 4/5 5 Listo

20 5/5 6 Listo

21 4/5 6 Listo

22 5/5 7 Listo

23 1/5 7 Listo

24 5/5 8 Listo

Marco Tamaño ID Estado

25 1/5 8 Listo

26 5/5 9 Listo

27 5/5 9 Listo

28 5/5 9 Listo

29 5/5 9 Listo

30 1/5 9 Listo

31 5/5 10 Listo

32 5/5 10 Listo

33 4/5 10 Listo

34 5/5 11 Listo

35 5/5 11 Listo

36 5/5 11 Listo

37 5/5 11 Listo

38 5/5 13 Listo

39 2/5 13 Listo

40 5/5 50 50

41 5/5 50 50

42 5/5 50 50

43 5/5 50 50

Ilustración 20 Ejemplo 2

Procesos suspendidos

Form ? X

TABLA DE PROCESOS

	ID	Operación	Resultado	Estado	TME	Llegada	Finalización	Retomo	Respuesta	Servicio	Espera	Restante
34	47	-271MOD104	Null	Nuevo	9	Null	Null	Null	Null	Null	Null	9
35	48	42*-335	Null	Nuevo	9	Null	Null	Null	Null	Null	Null	9
36	49	909+34	Null	Nuevo	6	Null	Null	Null	Null	Null	Null	6
37	50	502+657	Null	Nuevo	9	Null	Null	Null	Null	Null	Null	9
38	6	161-390	Null	Listo	10	0	Null	Null	Null	0	23	10
39	7	340-724	Null	Listo	8	0	Null	Null	Null	0	23	8
40	8	391+468	Null	Listo	8	0	Null	Null	Null	0	23	8
41	9	-368-981	Null	Listo	15	0	Null	Null	Null	0	23	15
42	10	-74*-384	Null	Listo	13	0	Null	Null	Null	0	23	13
43	11	367*966	Null	Listo	9	0	Null	Null	Null	0	23	9
44	12	-331-588	Null	Listo	8	6	Null	Null	Null	0	17	8
45	13	-829/-312	Null	Listo	8	7	Null	Null	Null	0	16	8
46	4	356+493	Null	Listo	10	0	Null	Null	12	8	15	2
47	3	-808/287	Null	Listo	9	0	Null	Null	5	7	16	2
48	1	81+449	Null	Suspendido	11	0	Null	Null	0	3	20	8
49	2	-453MOD877	Null	Suspendido	16	0	Null	Null	3	2	21	14
50	5	747MOD291	Null	Ejecución	7	0	Null	Null	20	3	20	4

Ilustración 21 BCP

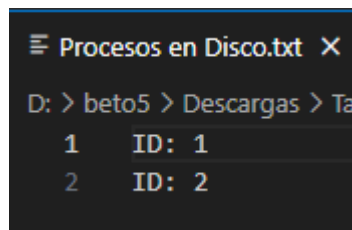


Ilustración 22 Txt

Enlace de descarga (contenido):

<https://drive.google.com/drive/folders/1JJmsWcGmNNmJOWiiBmLrGOEYfYcS-sRz?usp=sharing>

Conclusión

Lomeli Jaime:

Para finalizar, coincido en que esta actividad ha sido más sencilla que el reto que fue con paginación simple, el uso de archivo txt en Python no es tan complejo y nos permitió manejar los datos de manera rápida, pero útil. La mayor problemática que encontramos era relacionada con las teclas, ya que por algún motivo el presionar “spacebar” generaba que el programa fallara y se cerrara, fuera de ello solo fue necesario agregar en la tabla de procesos los suspendidos y el funcionamiento de estos mismos.

Monjaraz Luis:

En conclusión, esta actividad nos ayudo a poner en prueba nuestros conocimientos sobre la suspensión en la paginación simple, debo de mencionar, que esta actividad ha sido de las mas sencillas de realizar, pues como lo fue Round Robin en su momento solo teníamos que agregar nuevas teclas y una que otra función, las cuales ya teníamos echas con anterioridad desde el anterior programa. Por lo que sí, fue mas sencillo de realizar, aunque eso no evita que haya sido un reto el arreglar el motivo por el que llegaba a “tronar” el programa, siendo en este caso la barra espaciadora y el tabulador, esto por que al pulsar estas teclas se reinicia todo el proceso y no esta preparado para eso. La lectura de archivos también fue bastante útil a nivel conocimiento, pues en lo personal solo la conocía en C++ siendo que la verdad en Python es mucho más sencilla su implementación.