Assignments

Assignment 4 - In progress

Honor Pledge Accepted
Draft - In progress
Submitted
Returned

field[i] => char*[20] => array of pointers.
of char.

=> char *str[i].

Assignment Details

Title
Assignment 4
Due
Mar 21, 2023 11:55 PM

20

Number of resubmissions allowed
Unlimited
Accept Resubmission Until
Mar 23, 2023 11:55 PM
Status
Honor Pledge Accepted
Grade Scale
Points (max 100.00)
Modified by instructor
Mar 8, 2023 9:25 PM

Instructions

Assignment overview

As discussed in class, there are multiple CPU scheduling algorithms that could be used for a given set of processes. There are several methods one could employ to evaluate which CPU scheduling algorithm is best. Deterministic modeling is one type of analytic evaluation. This method takes a particular predetermined workload and defines the performance of each algorithm for that workload. We employed this method **manually** in class to determine the wait and turnaround times for a set of processes with a given arrival time and burst time.

For this assignment, write a C program which will **automatically** simulate multiple CPU scheduling algorithms for any given set of processes. The program will maintain wait and turnaround times for each process over time and then display the average wait and turnaround time for all processes.

Purpose

The goals of this assignment are the following:

- Learn the differences between various CPU scheduling algorithms and how those differences influence average wait and turnaround times.

Computing platform

You are welcome to develop your program on your own workstation if you wish, but you are responsible for ensuring that your program compiles and runs without error on the Gaul computing platform. Marks will be deducted if your program fails to compile, or your program runs into errors on Gaul.

- https://wiki.sci.uwo.ca/sts/computer-science/gaul

Instructions

Attached to this assignment is a tarball with the following files in it. **None of these files should be modified**:

```
Makefile                  <--- A pre-packaged Makefile. This tells you how your program should be structured
run-assignment.sh         <--- A shell script that will automatically run your program
assignment-4-input.csv    <--- A comma separated list of processes and their burst times. One process per line. Each line represents the ar
```

Download this tarball and upload it to Gaul. Extract the tarball (`tar -xvf assignment-4.tar`). Change to the `assignment-4` directory.

You will write a program called `assignment-4.c`. This program will:

- Accept 2 or 3 parameters (You can assume parameters are supplied in this order)
  - The first parameter defines the algorithm to simulate.
    - -f for First Come First Served
    - -s for Shortest Job First. This should be implemented with preemption.
    - -r <integer> for Round Robin
      - (If -r is supplied, the next parameter is a positive integer defining the time quantum)
  - The next parameter defines the filename to read input from. For the purposes of this assignment, this will always be assignment-4-input.csv, but of course, you can create your own versions for testing
- Depending on which algorithm is selected, your program should display the simulation of that algorithm including each processes' wait and turnaround time for each "tick" (one tick per line).
- Once all processes have finished, display the average wait time and turnaround time for all processes using that algorithm (to an accuracy of one decimal point)

Output

Executing `./assignment-4 -f assignment-4-input.csv` should produce the following output:

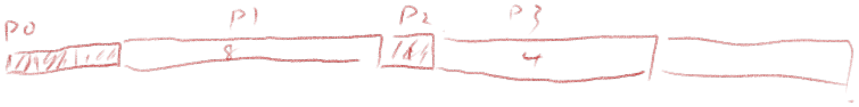
```
First Come First Served
T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
```

```
T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
T2 : P0  - Burst left  1, Wait time  0, Turnaround time  2
T3 : P1  - Burst left  8, Wait time  2, Turnaround time  2
T4 : P1  - Burst left  7, Wait time  2, Turnaround time  3
T5 : P1  - Burst left  6, Wait time  2, Turnaround time  4
T6 : P1  - Burst left  5, Wait time  2, Turnaround time  5
T7 : P1  - Burst left  4, Wait time  2, Turnaround time  6
T8 : P1  - Burst left  3, Wait time  2, Turnaround time  7
...

P18
        Average waiting time:         178
        Average turnaround time:      187

P19
        Average waiting time:         186
        Average turnaround time:      192

Total average waiting time:     84.6
Total average turnaround time:  95.1
```

turnAround.

Note that in this example, even though P1 arrived at T1, P0 had not finished yet. P0 runs until completion at the end of T2. Then at T3, P1 will begin its burst.

Executing `./assignment-4 -s assignment-4-input.csv` should produce the following output:



```
Shortest Job First
T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
T2 : P2  - Burst left  1, Wait time  0, Turnaround time  0
T3 : P0  - Burst left  1, Wait time  1, Turnaround time  3
T4 : P3  - Burst left  4, Wait time  1, Turnaround time  1
T5 : P3  - Burst left  3, Wait time  1, Turnaround time  2
T6 : P3  - Burst left  2, Wait time  1, Turnaround time  3
T7 : P3  - Burst left  1, Wait time  1, Turnaround time  4
T8 : P8  - Burst left  3, Wait time  0, Turnaround time  0
...
P18
        Waiting time:           8
        Turnaround time:       17

P19
        Waiting time:           1
        Turnaround time:        7

Total average waiting time:     58.8
Total average turnaround time:  69.3
```

Note that in this example, at T2, even though P0 still has a burst left of 2, P2 has arrived and has a burst left of 1. Therefore, P0 is preempted, P2 runs its CPU burst. At T3, P0 is the shortest process and therefore finishes its last CPU burst. Then P3 runs its CPU burst, and so on.

Executing `./assignment-4 -r 3 assignment-4-input.csv` should produce the following output:

```
Round Robin with Quantum 3
T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
T2 : P0  - Burst left  1, Wait time  0, Turnaround time  2
T3 : P1  - Burst left  8, Wait time  2, Turnaround time  2
T4 : P1  - Burst left  7, Wait time  2, Turnaround time  3
T5 : P1  - Burst left  6, Wait time  2, Turnaround time  4
T6 : P2  - Burst left  1, Wait time  4, Turnaround time  4
T7 : P3  - Burst left  4, Wait time  4, Turnaround time  4
T8 : P3  - Burst left  3, Wait time  4, Turnaround time  5
T9 : P3  - Burst left  2, Wait time  4, Turnaround time  6
T10: P4  - Burst left 12, Wait time  6, Turnaround time  6
...
P18
        Waiting time:         116
        Turnaround time:      125

P19
        Waiting time:          77
        Turnaround time:       83

Total average waiting time:     118.9
Total average turnaround time:  129.4
```

if p.time > Q { p.time -= Q

ttime += Q

Q: 3

time	0	1	2	3	4	5	6	7	8	9	10	11
task	0	0	0	1	1	1	2	2	2	0	0	0

$(time \% 9) / 3$.

if task is done:

Note that P2 was able to finish its CPU burst at T6 which was less than the time quantum of 3. Therefore, a context switch to P3 occurs and the time quantum is reset to 3. P3 is then awarded a time quantum 3.

Executing `./assignment-4 -r 12 assignment-4-input.csv` should produce the following output:

```
Round Robin with Quantum 12
T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
T2 : P0  - Burst left  1, Wait time  0, Turnaround time  2
T3 : P1  - Burst left  8, Wait time  2, Turnaround time  2
T4 : P1  - Burst left  7, Wait time  2, Turnaround time  3
T5 : P1  - Burst left  6, Wait time  2, Turnaround time  4
T6 : P1  - Burst left  5, Wait time  2, Turnaround time  5
T7 : P1  - Burst left  4, Wait time  2, Turnaround time  6
T8 : P1  - Burst left  3, Wait time  2, Turnaround time  7
...
P18
        Waiting time:         144
        Turnaround time:      153

P19
        Waiting time:         152
        Turnaround time:      158

Total average waiting time:     112.8
Total average turnaround time:  123.4
```

```
                  ↓
 0   1   2   3   4   5
| 2 | 0 | 1 | 1 | 0 | 0 |

for(int i = index. ; i < len(arr) ; i++) {
    if i == len(arr) -1    i = 0 ;

    while arr[i] == 0 :
        i++

    return i;
```

and `./run-assignment.sh 4` should produce the following output:

```
Assignment 4 STARTED - Dow Mon  ## ##:##:## AM/PM EST 2023

Cleaning environment
----------------------------
rm -f assignment-4

Checking environment
----------------------------
c349fde53687117c81315563aedb1c09  ./run-assignment.sh
<unique id>  assignment-4.c
Makefile: OK
assignment-4-input.csv: OK

Building environment
----------------------------
make all
make[1]: Entering directory '/home/wbeldman/3305/Projects/Assignment 4'
gcc -o assignment-4 assignment-4.c -Wall -Wpedantic -Wextra -std=gnu17
make[1]: Leaving directory '/home/wbeldman/3305/Projects/Assignment 4'

Assignment 4
----------------------------

-- Invalid parameters (0) --
Proper usage is ./assignment-4 [-f|-s|-r <quantum>] <Input file name>

-- Invalid parameters (4+) --
Proper usage is ./assignment-4 [-f|-s|-r <quantum>] <Input file name>

-- Invalid filename --
First Come First Served
Could not open file-does-not-exist.txt

-- First Come First Served --
First Come First Served
First Come First Served
T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
T2 : P0  - Burst left  1, Wait time  0, Turnaround time  2
T3 : P1  - Burst left  8, Wait time  2, Turnaround time  2
T4 : P1  - Burst left  7, Wait time  2, Turnaround time  3
T5 : P1  - Burst left  6, Wait time  2, Turnaround time  4
T6 : P1  - Burst left  5, Wait time  2, Turnaround time  5
T7 : P1  - Burst left  4, Wait time  2, Turnaround time  6
T8 : P1  - Burst left  3, Wait time  2, Turnaround time  7
T9 : P1  - Burst left  2, Wait time  2, Turnaround time  8
T10: P1  - Burst left  1, Wait time  2, Turnaround time  9
T11: P2  - Burst left  1, Wait time  9, Turnaround time  9
T12: P3  - Burst left  4, Wait time  9, Turnaround time  9
...
T209: P19 - Burst left  2, Wait time 186, Turnaround time 190
T210: P19 - Burst left  1, Wait time 186, Turnaround time 191

P0
        Waiting time:         0
        Turnaround time:        3

P1
        Waiting time:           2
        Turnaround time:       10
...
P18
        Waiting time:         178
        Turnaround time:      187

P19
        Waiting time:         186
        Turnaround time:      192

Total average waiting time:     84.6
Total average turnaround time:  95.1

-- Shortest Job First --
Shortest Job First
T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
T2 : P2  - Burst left  1, Wait time  0, Turnaround time  0
T3 : P0  - Burst left  1, Wait time  1, Turnaround time  3
T4 : P3  - Burst left  4, Wait time  1, Turnaround time  1
T5 : P3  - Burst left  3, Wait time  1, Turnaround time  2
T6 : P3  - Burst left  2, Wait time  1, Turnaround time  3
T7 : P3  - Burst left  1, Wait time  1, Turnaround time  4
T8 : P8  - Burst left  3, Wait time  0, Turnaround time  0
T9 : P8  - Burst left  2, Wait time  0, Turnaround time  1
T10: P8  - Burst left  1, Wait time  0, Turnaround time  2
T11: P1  - Burst left  8, Wait time 10, Turnaround time 10
T12: P1  - Burst left  7, Wait time 10, Turnaround time 11
...
T209: P5  - Burst left  2, Wait time 187, Turnaround time 204
T210: P5  - Burst left  1, Wait time 187, Turnaround time 205

P0
        Waiting time:           1
        Turnaround time:        4

P1
        Waiting time:          11
        Turnaround time:       19
...
P18
        Waiting time:           8
        Turnaround time:       17

P19
        Waiting time:           1
        Turnaround time:        7

Total average waiting time:     58.8
Total average turnaround time:  69.3

-- Round Robin (Time Quantum 3) --
Round Robin with Quantum 3
T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
T2 : P0  - Burst left  1, Wait time  0, Turnaround time  2
T3 : P1  - Burst left  8, Wait time  2, Turnaround time  2
T4 : P1  - Burst left  7, Wait time  2, Turnaround time  3
T5 : P1  - Burst left  6, Wait time  2, Turnaround time  4
T6 : P2  - Burst left  1, Wait time  4, Turnaround time  4
T7 : P3  - Burst left  4, Wait time  4, Turnaround time  4
T8 : P3  - Burst left  3, Wait time  4, Turnaround time  5
T9 : P3  - Burst left  2, Wait time  4, Turnaround time  6
T10: P4  - Burst left 12, Wait time  6, Turnaround time  6
T11: P4  - Burst left 11, Wait time  6, Turnaround time  7
T12: P4  - Burst left 10, Wait time  6, Turnaround time  8
...
T209: P15 - Burst left  1, Wait time 179, Turnaround time 194
T210: P5  - Burst left  1, Wait time 187, Turnaround time 205

P0
        Waiting time:           0
        Turnaround time:        3
```

```
P1
        Waiting time:          95
        Turnaround time:      103
...
P18
        Waiting time:         116
        Turnaround time:      125

P19
        Waiting time:          77
        Turnaround time:       83

Total average waiting time:     118.9
Total average turnaround time:  129.4

-- Round Robin (Time Quantum 12) --
Round Robin with Quantum 12
T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
T2 : P0  - Burst left  1, Wait time  0, Turnaround time  2
T3 : P1  - Burst left  8, Wait time  2, Turnaround time  2
T4 : P1  - Burst left  7, Wait time  2, Turnaround time  3
T5 : P1  - Burst left  6, Wait time  2, Turnaround time  4
T6 : P1  - Burst left  5, Wait time  2, Turnaround time  5
T7 : P1  - Burst left  4, Wait time  2, Turnaround time  6
T8 : P1  - Burst left  3, Wait time  2, Turnaround time  7
T9 : P1  - Burst left  2, Wait time  2, Turnaround time  8
T10: P1  - Burst left  1, Wait time  2, Turnaround time  9
T11: P2  - Burst left  1, Wait time  9, Turnaround time  9
T12: P3  - Burst left  4, Wait time  9, Turnaround time  9
...
T209: P15 - Burst left  1, Wait time 179, Turnaround time 194
T210: P17 - Burst left  1, Wait time 181, Turnaround time 193

P0
        Waiting time:           0
        Turnaround time:        3

P1
        Waiting time:           2
        Turnaround time:       10
...

P18
        Waiting time:         144
        Turnaround time:      153

P19
        Waiting time:         152
        Turnaround time:      158

Total average waiting time:     112.8
Total average turnaround time:  123.4

Cleaning environment
----------------------------
rm -f assignment-4

Assignment 4 COMPLETED - Dow Mon ## ##:##:## AM/PM EST 2023
```

Helpful hints

- Your program should read in all the data and populate the data into some kind of data structure. An array of structures or a multi-dimensional array would be the best way to represent the data. After you have read the entire input file and have the data, you run a loop to step through all the processes until the CPU burst for all processes is 0.
- Each process will need to retain the **arrival time**, the **burst time**, the **wait time**, and the **turnaround time**. At every tick, you must make some updates on **all** processes, not just the active process.
    - The arrival time is easily determined by the process read which is the same as the line read. For example, the arrival time for P0 is 0. The arrival time for P4 is 4. This value never gets updated.
    - The burst time will start at the value read in. It will be decremented every time it is the **active** process.
    - The wait time will start at 0. It will be incremented every time the current time is greater than or equal to the arrival time, the burst time is not 0, and it is **not the active** process.
    - The turnaround time will start at 0. It will be incremented every time the current time is greater than or equal to the arrival time, and the burst time is not 0. This is irrespective of whether this is the active process or not.
- It's important to remember that each tick represents the start of that tick. For example:

    -
      ```
      T0 : P0  - Burst left  3, Wait time  0, Turnaround time  0
      T1 : P0  - Burst left  2, Wait time  0, Turnaround time  1
      ```

```
T2 : P0  - Burst left  1, Wait time  0, Turnaround time  2
T3 : P1  - Burst left  8, Wait time  2, Turnaround time  2
```

  - This means P0 started at T0 and finished at the start of T3 (represented as 0-3 in our Gantt charts). So its turnaround time is 3, although this is not actually represented on the screen. P1 arrived at T1 and did not start until T3. Therefore, its wait time is 3-1 = 2. Its wait time and turnaround time is at least 2. Every time tick that P1 is active, the turnaround time is incremented and the burst time is decremented. Every time tick that P1 is not active and not complete, the wait time and the turn around time is incremented.
- Here is some sample code for reading in a file line-by-line
  - https://www.geeksforgeeks.org/scanf-and-fscanf-in-c/

Submitting

When you are finished your assignment, follow these steps

1. From inside the assignment-4 directory, run the following command (make sure `run-assignment.sh` is executable): `script -c './run-assignment.sh 4' assignment-4.out`
   Your directory should now contain the following files:

```
assignment-4.c    <--- Your program
assignment-4.out  <--- The output produced by running the script command above
Makefile          <--- A pre-packaged Makefile. This tells you how your program should be structured
run-assignment.sh <--- A shell script that will automatically run your program and put the results in assignment-4.out
assignment-4-input.csv <--- A comma separated list of processes and their burst times. One process per line. Each line represents the arriv
```

2. Assuming the command was successful, run the follow command to get out of the `assignment-4` directory: `cd ..`
3. Package your assignment into a tarball: `tar -cvf assignment-4.tar assignment-4`
4. Verify the contents of your tarball (`tar -tvf assignment-4.tar`) (`du -sh assignment-4.tar`). **If your tarball is 10kb in size** you have an empty tarball and you made an error on this step. Make sure you are properly creating your tarball with the right files in it.
5. Use an SFTP program to download the tarball and then upload it to OWL.

Additional resources for assignment

- Assignment-4.tar (10 KB; Feb 21, 2023 9:29 pm)

Grading Rubric

Preview Rubric

Submission

Attachments

No attachments yet

Select a file from computer Choose File No file chosen

Proceed Preview Save Draft Cancel ❗ Don't forget to save or proceed!