

دستور کار آزمایشگاه سیستم های عامل



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

آزمایشگاه سیستم های عامل

آرین محسنی

کاوه احمدی

آزمایش ۸

آذر 1403

دستور کار آزمایشگاه سیستم های عامل

در این آزمایش میخواهیم الگوریتم های زمان بندی را پیاده سازی کنیم.
به دلخواه دو الگوریتم SJF و Round-Robin را پیاده سازی میکنیم.

کد های مربوط به این دو الگوریتم در زیر آمده اند.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  void swap(int* xp, int* yp){
6      int temp = *xp;
7      *xp = *yp;
8      *yp = temp;
9  }
10
11 void bubbleSort(int arr[], int n){
12     int i, j;
13     bool swapped;
14     for (i = 0; i < n - 1; i++) {
15         swapped = false;
16         for (j = 0; j < n - i - 1; j++) {
17             if (arr[j] > arr[j + 1]) {
18                 swap(&arr[j], &arr[j + 1]);
19                 swapped = true;
20             }
21         }
22         if (swapped == false)
23             break;
24     }
25 }
26
27 int main(){
28     int process_count;
29     scanf("%d",&process_count);
30
31     int CPUburst[process_count];
32     int waitTime[process_count];
33     int cpuburst;
34
35     for (int i = 0; i < process_count; i++)
36     {
37         scanf("%d",&cpuburst);
38
39         CPUburst[i] = cpuburst;
40         waitTime[i] = 0;
41         //CPUburst[0] = 0;
42     }
43
44     bubbleSort(CPUburst,process_count);
45     for (int i = 1; i < process_count; i++)
46     {
47         waitTime[i] = waitTime[i-1] + CPUburst[i-1];
48     }
49
50     int sum = 0;
51     printf("-----");
52
53     //test print
54     for (int i = 0; i < process_count; i++)
55     {
56         //printf("%d",CPUburst[i]);
57         sum += waitTime[i];
58         printf("\n%d, WaitTime:%d\n",i+1,waitTime[i]);
59     }
60
61     double averageWaiting = (double)sum/process_count;
62
63     printf("-----");
64     printf("\nAverage :%lf\n",averageWaiting);
65
66     return 0;
67 }
```

دستور کار آزمایشگاه سیستم های عامل

مثال برای تست:

```
4
1
2
3
4
-----
1, WaitTime:0

2, WaitTime:1

3, WaitTime:3

4, WaitTime:6
-----
Average :2.500000
```

حتی اگر ترتیب ورودی نیز فرق کند، تاثیری ندارد

```
4
3
4
2
1
-----
1, WaitTime:0

2, WaitTime:1

3, WaitTime:3

4, WaitTime:6
-----
Average :2.500000
```

دستور کار آزمایشگاه سیستم های عامل

در بخش بعدی الگوریتم Round-Robin را پیاده سازی کرده ایم که با دادن پردازش ها و کوانتوم زمانی، زمانبندی انجام میشود.

```
1  #include <stdio.h>
2
3  void findWaitingTime(int processes[], int n, int bt[], int wt[], int tq) {
4      int rem_bt[n];
5      for (int i = 0; i < n; i++) {
6          rem_bt[i] = bt[i];
7      }
8
9      int t = 0; // Time variable
10     while (1) {
11         int done = 1;
12         for (int i = 0; i < n; i++) {
13             if (rem_bt[i] > 0) {
14                 done = 0;
15                 if (rem_bt[i] > tq) {
16                     t += tq;
17                     rem_bt[i] -= tq;
18                 } else {
19                     t += rem_bt[i];
20                     wt[i] = t - bt[i];
21                     rem_bt[i] = 0;
22                 }
23             }
24         }
25         if (done == 1) {
26             break;
27         }
28     }
29 }
30
31 void findTurnaroundTime(int processes[], int n, int bt[], int wt[], int tat[]) {
32     for (int i = 0; i < n; i++) {
33         tat[i] = bt[i] + wt[i];
34     }
35 }
36
37 void findAverageTime(int processes[], int n, int bt[], int tq) {
38     int wt[n], tat[n];
39     findWaitingTime(processes, n, bt, wt, tq);
40     findTurnaroundTime(processes, n, bt, wt, tat);
41
42     int total_wt = 0, total_tat = 0;
43     printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");
44
45     for (int i = 0; i < n; i++) {
46         total_wt += wt[i];
47         total_tat += tat[i];
48         printf("%d\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);
49     }
50
51     printf("\nAverage waiting time: %.2f\n", (float)total_wt / n);
52     printf("Average turnaround time: %.2f\n", (float)total_tat / n);
53 }
54
55 int main() {
56     int n, tq;
57     printf("Enter the number of processes: ");
58     scanf("%d", &n);
59
60     int processes[n], bt[n];
61     printf("Enter the burst times of the processes: \n");
62     for (int i = 0; i < n; i++) {
63         printf("Process %d: ", i + 1);
64         scanf("%d", &bt[i]);
65         processes[i] = i + 1;
66     }
67 }
```

دستور کار آزمایشگاه سیستم های عامل

مثال:

```
Enter the number of processes: 5
Enter the burst times of the processes:
Process 1: 1
Process 2: 6
Process 3: 2
Process 4: 5
Process 5: 8
Enter the time quantum: 3
Process Burst Time      Waiting Time      Turnaround Time
1         1              0                 1
2         6              9                 15
3         2              4                 6
4         5              12                17
5         8              14                 22

Average waiting time: 7.80
Average turnaround time: 12.20
```

بخش دوم

بخش دوم دستور کار که مرتبط با CFS است به صورت کامل در فایل زیپ آورده شده است. توضیحات روند آن مانند دستور کار آزمایشگاه میباشد.