VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B.Tech. Winter Semester 2023-24
School Of Computer Science and Engineering (SCOPE)

Digital Assignment - II

Operating System Lab

Apurva Mishra, 22BCE2791

16 August 2024

1. Questions

Problem 1.1.
Write a LINUX/UNIX C Program for the Implementation of First Come First Serve Scheduling Algorithm.

```c
#include <stdio.h>
#include <stdlib.h>

struct job {
  int uuid;
  int time; // burst time
};

struct job job_new(int uuid, int time) {
  struct job job;
  job.uuid = uuid;
  job.time = time;

  return job;
}

struct queue {
  int capacity;
  int length;
  struct job *jobs;
};

struct queue *queue_new() {
  struct queue *queue = malloc(sizeof(struct queue));
  struct job *jobs = malloc(sizeof(struct job) * 10);
  queue->capacity = 5;
  queue->length = 0;
  queue->jobs = jobs;

  return queue;
}

bool queue_is_empty(struct queue *queue) {
  if (queue->length <= 0) {
    return true;
  }
  return false;
}

void increase_capacity(struct queue *queue) {
  struct job *new_jobs = malloc(sizeof(struct job) * (queue->capacity + 5));
  for (int i = 0; i < queue->length; i++) {
    new_jobs[i] = queue->jobs[i];
  }

  queue->capacity += 5;

  free(queue->jobs);
  queue->jobs = new_jobs;
}
```

```c
void queue_add_job(struct queue *queue, struct job job) {
  if (queue->length == queue->capacity) {
    increase_capacity(queue);
  }
  queue->jobs[queue->length] = job;
  queue->length += 1;
}

void input_jobs(struct queue *queue) {
  int n_job;
  printf("Enter total number of processes:\n");
  scanf("%d", &n_job);

  printf("Enter Process Burst Time:\n");
  for (int i = 0; i < n_job; i++) {
    int burst_time;
    printf("P[%d]:", i + 1);
    scanf("%d", &burst_time);
    queue_add_job(queue, job_new(i + 1, burst_time));
  }
}

int queue_process(struct queue *queue) {
  int total = 0;
  printf("Process Burst_Time Waiting_Time Turnaround_Time\n");
  fflush(stdout);
  for (int i = 0; i < queue->length; i++) {
    printf("%6d %10d %12d %15d\n ", queue->jobs[i].uuid, queue->jobs[i].time,
           total, queue->jobs[i].time + total);

    total += queue->jobs[i].time;
  }

  printf("\nTotal Time: %d\n", total);
  printf("Average waiting time: %d\n", total / queue->length);

  return total;
}

int main() {
  struct queue *queue = queue_new();

  input_jobs(queue);

  queue_process(queue);

  free(queue->jobs);
  free(queue);
  return 0;
}
```

Output

```
college/os/ass2 via C v15.0.0-clang via t v0.11.1
❯ who; date now;
apurva           console      Aug  2 21:51
apurva           ttys000      Aug 15 14:09
Fri, 16 Aug 2024 18:32:31 +0530 (now)

college/os/ass2 via C v15.0.0-clang via t v0.11.1
❯ just r
zig cc ./main.c -o main --std=c23
./main
Enter total number of processes:
6
Enter Process Burst Time:
P[1]:12
P[2]:4
P[3]:1
P[4]:34
P[5]:18
P[6]:4
Process Burst_Time Waiting_Time Turnaround_Time
     1         12            0              12
      2          4           12              16
      3          1           16              17
      4         34           17              51
      5         18           51              69
      6          4           69              73

Total Time: 73
Average waiting time: 12
```

Problem 1.2.
Write a shell script program that uses * and number (1 - 4) to print the following pattern (shown below). To print the left and right parts of the pattern, use nested loops.

```
      *
    * * *
  * * * * *
* * * * * * *
1 1 1 1 1 1 1
  2 2 2 2 2
    3 3 3
      4
```

```bash
#!/bin/bash

print_stars() {
    for ((i=0; i<=3; i++)); do
        for ((j=1; j<=(3-i); j++)); do
            echo -n " "
        done
        for ((j=1; j<=(2*i+1); j++)); do
            echo -n "*"
        done
        echo
    done
}

print_numbers() {
    for ((i=3; i>=0; i--)); do
        for ((j=1; j<=(3-i); j++)); do
            echo -n " "
        done
        for ((j=1; j<=(2*i+1); j++)); do
            tmp=$((4-i))
            echo -n "$tmp"
        done
        echo
    done
}

print_stars
print_numbers
```

```
college/os/ass2 via t v0.11.1
❯ who; date now;
apurva           console      Aug  2 21:51
apurva           ttys000      Aug 15 14:09
Fri, 16 Aug 2024 15:14:36 +0530 (now)

college/os/ass2 via t v0.11.1
❯ bash q2.sh
   *
  ***
 *****
*******
1111111
 22222
  333
   4
```