

ADVANCED UNIX PROGRAMMING MIDTERM REPORT



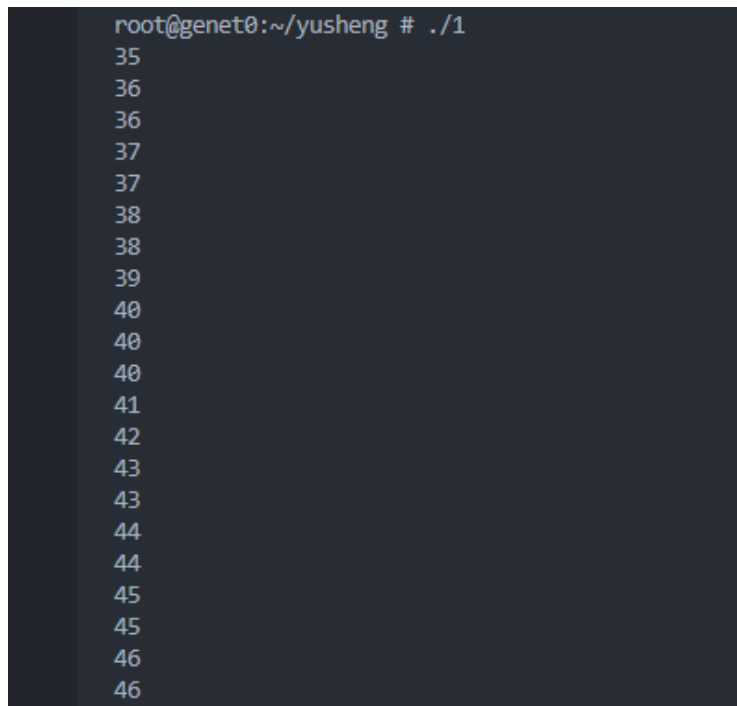
TEAM 9 — 林禾堃、馬毓昇、陳曦

Dec 2023

1. Question 1

How to run the code: `./q1`

1.1 Result explanation



```
root@genet0:~/yusheng # ./1
35
36
36
37
37
38
38
39
40
40
40
41
42
43
43
44
44
45
45
46
46
```

Above is a partial result of q1.c. As you can see, although we let the program call multiple `sleep(10)`, and print the second data every six calls (every minute), the output value is increasing little by little. The reason is that every call to `sleep()` will take some additional system CPU time other than the 10 second itself, resulting in an accumulation of tiny delays.

1.2 Answer to the question

A program like cron daemon will not encounter accumulating delays, this is because they don't rely on `sleep()` to decide when to run, but decide it by the system clock, avoiding this issue.

2. Question 2

How to run the code: `./q2 <path>`

3. Question 3

How to run the code: `./q3`

4. Question 4

How to run the code: `./q4`

4.1 Code implementation

Originally, the new nodes were statically created. Since this will cause memory problems, we modified it:

```
struct ListNode* append_node(struct ListNode* head, int val){
    if(head == NULL){
        struct ListNode *newNode = (struct ListNode*)malloc(sizeof(struct ListNode));
        newNode->val = val;
        newNode->next = NULL;
        head = newNode;
        return head;
    }else {
        struct ListNode *newNode = (struct ListNode*)malloc(sizeof(struct ListNode));
        struct ListNode *tail = head;
        while(tail->next != NULL)tail = tail->next;
        newNode->val = val;
        tail->next = newNode;
        newNode->next = NULL;
    }
    return head;
}
```

Now, new nodes are created while allocating memory for the pointer `newNode` with `malloc()`.

```
    struct ListNode *node = head, *tmp;
    while(node != NULL){
        printf("%d\n", node->val);
        tmp = node;
        node = node->next;
        free(tmp);
    }
    return 0;
}
```

And since we used malloc to allocate memory, at this session in main, we added some lines to free the memory after printing the outputs.