

Bài tập chương 4

Hà Lê Hoài Trung

Bài tập 1

Trong các giải thuật sau, giải thuật nào có thể gây ra trường hợp 1 process có thể không bao giờ được thực thi:

1. First come, first serve
2. Shortest job first
3. Round robin
4. Priority.

Giải thích lý do tại sao xảy ra trường hợp trên?

Bài tập 2

- Xét 1 tập các process sau có thời gian thực thi CPU tính bằng mili giây:

Process	Arrival- time	Burst - time
P1	0	7
P2	2	2
P3	4	5
P4	7	3
p5	9	6

- Vẽ sơ đồ Gantt thực thi của các process theo giải thuật định thời: FCFS, SJF, RR (quantum = 3), cùng độ ưu tiên xét P_i và P_j , P_i ưu tiên hơn nếu $i < j$.
- Thời gian chờ, thời gian đáp ứng, thời gian hoàn thành của các process trong từng giải thuật định thời?

Bài tập 3

- Xét 1 tập các process sau có thời gian thực thi CPU tính bằng mili giây:

Process	Arrival- time	Burst - time	Priority
P1	0	5	3
P2	2	2	2
P3	4	8	1
P4	7	3	5
p5	9	4	4

- Vẽ sơ đồ Gantt thực thi của các process theo giải thuật định thời: FCFS, SRTF, RR (quantum = 3), Priority (thực hiện chế độ preemptive) (càng nhỏ càng ưu tiên, cùng độ ưu tiên xét P_i và P_j , P_i ưu tiên hơn nếu $i < j$).
- Thời gian chờ, thời gian đáp ứng, thời gian hoàn thành của các process trong từng giải thuật định thời?

Bài tập 4

- Cho 4 tiến trình A, B, C, D với thời gian vào ready list và thời gian cần CPU cho các lần thứ 1, thứ 2, thứ 3 và thời gian thực hiện I/O tương ứng như bản sau:

Process	Arrival time	1 st exec	1 st I/O	2 nd exec	2 nd I/O	3 rd exec
A	0	4	4	4	4	4
B	2	8	1	8	-	-
C	3	2	1	2	-	-
D	7	1	1	1	1	1

- Vẽ giản Gantt, Tính thời gian đợi trung bình, Thời gian đáp ứng trung bình, Thời gian lưu lại trong hệ thống trung bình cho các giải thuật.
 1. FCFS
 2. RR với $q = 3$
 3. SRFT.

Bài tập 5

- Cho 4 tiến trình A, B, C, D với thời gian vào ready list và thời gian cần CPU cho các lần thứ 1, thứ 2, thứ 3 và thời gian thực hiện I/O tương ứng như bản sau:

Process	Arrival time	1 st exec	1 st I/O	2 nd exec	2 nd I/O
A	0	3	1	3	1
B	2	5	2	-	-
C	5	2	3	3	-
D	7	6	2	5	3

- Vẽ giản Gantt, Tính thời gian đợi trung bình, Thời gian đáp ứng trung bình, Thời gian lưu lại trong hệ thống trung bình cho các giải thuật.
 1. FCFS
 2. RR với $q = 3$
 3. SRFT

Bài tập 6

Process	Arrival time	Burst time (ms)	Priority
P1	0	8	3
P2	1	4	2
P3	2	3	1
P4	4	5	4

1. RR với $q = 2$
2. Preemptive Priority với số càng lớn càng ưu tiên
3. Điều phối ưu tiên nhiều cấp xoay vòng, sử dụng 2 cấp: Cấp 1 sử dụng giải thuật robin round với quantumn = 3ms. Cấp 2 sử dụng giải thuật SRTF. Một process nếu đã ở cấp I 5ms sẽ được chuyển xuống cấp II nếu đang ở trạng thái waiting còn nếu đang ở trạng thái running thì sau khi ra khỏi sẽ chuyển. Ngược lại một process đang ở cấp II sau khoảng thời gian 10ms sẽ được chuyển lên I. Khi các process vào bộ nhớ chính thì điều vào hàng đợi cấp I.

Bài tập 7

Process	Arrival time (ms)	Burst time (ms)
P1	0	8
P2	2	5
P3	4	7
P4	7	3

Điều phối ưu tiên nhiều cấp xoay vòng, sử dụng 2 cấp: Cấp 1 sử dụng giải thuật robin round với quantum = 3ms. Cấp 2 sử dụng giải thuật SRTF. Một process nếu đã ở cấp I 5ms sẽ được chuyển xuống cấp II nếu đang ở trạng thái waiting còn nếu đang ở trạng thái running thì sau khi ra khỏi sẽ chuyển. Ngược lại một process đang ở cấp II sau khoảng thời gian 10ms sẽ được chuyển lên I. Khi các process vào bộ nhớ chính thì điều vào hàng đợi cấp I.

Bài tập 8

```
/* test.c */
int main(int argc, char** argv)
{
    printf("Hello world.\n");
    printf("hi OS.\n");
    exit(0);
}
```

Biên dịch chương trình
trong Linux
gcc test.c -o test

Thực thi chương trình test
./test

Trong hệ thống sẽ có một
quá trình *test* được tạo ra,
thực thi và kết thúc.

- Liệt kê chuỗi trạng thái của quá trình test trong trường hợp tốt nhất:

Bài tập 9

```
/* test.c */
int main(int argc, char** argv)
{
    int n = 5;
    char* str;
    if (n % 2 == 0)
        str = "số chẵn \n";
    else
        str = "số lẻ \n";
    printf(str);
    exit(0);
}
```

Biên dịch chương trình
trong Linux
gcc test.c -o test

Thực thi chương trình test
./test

Trong hệ thống sẽ có một
quá trình *test* được tạo ra,
thực thi và kết thúc.

- Liệt kê chuỗi trạng thái của quá trình test trong trường hợp tốt nhất:

Bài tập 10

```
void main(){  
    printf ("hi");  
    fork ();  
    fork ();  
    printf ("Hello");  
    fork ();  
    printf ("Bye");  
}
```

- Hỏi chương trình in ra các dòng chữ nào trên màn hình.
- Hỏi sau khi thực hiện đoạn lệnh trên có bao nhiêu process.

Bài tập 11

```
void main() {  
    int pid;           1  
    printf ("hi");     2  
    pid = fork ();     3  
    if( pid == 0) {    4  
        fork ();      5  
        printf ("child"); 6  
    } else if (pid > 0) { 7  
        fork();        8  
        fork();        9  
        printf(parent); 10  
    } else             11  
        printf ("Bye"); 12  
}
```

- Hỏi chương trình in ra các dòng chữ nào trên màn hình.
- Có bao nhiêu process sau khi thực hiện chương trình trên.

Bài tập