

Subject: Lecture on Operating System Lecture Exercise 4.3

From: IKENOYA Katsutoshi <j05002@ie.u-ryukyu.ac.jp>

Date: Mon, 15 Jan 2007 19:43:24 +0900

To: Shinji KONO <kono@ie.u-ryukyu.ac.jp>

学籍番号 : 055702B

問題4.3

main.c を変更して、priority scheduling と round robin scheduling をシミュレーションせよ。

以下のデータを用いてシミュレートした。

```
- task_data.03
n 0 a 160 1
n 0 b 80 3
n 0 c 60 2
n 0 d 10 5
n 0 e 50 4
```

(1)priority scheduling

- ・ソースの抜粋

ソースの場所 : /home/y05/j05002/os-queue/main.c

```
void
pri(QueuePtr task)
{
QueuePtr q;
task = sort_queue(task,by_priority);
while(task) {
task = dequeue(task,&q);
exec_task(q,quantum);
}
}
```

- ・実行結果

```
[j05002@os-queue]% ./os-queue < data/task_data.03
0 a: 160
160 c: 60
220 b: 80
300 e: 50
350 d: 10
360 end:
```

```
average waiting time = 206
average turn around time = 278
```

priorityの高いものから実行されていることが確認できる。

(2)round robin scheduling

- ・ソースの抜粋

ソースの場所 : /home/y05/j05002/os-queue/main.c

```
static int prev = 0;
```

```

void
rr(QueuePtr task)
{
// 再帰処理のフラグ
int r_flag = 1;
QueuePtr q,task_save;
task = sort_queue(task,by_priority);

while(r_flag == 1){
r_flag = 0;
task_save = task;

while(task) {
task = dequeue(task,&q);

// 既に終了したプロセスを実行しないための判別処理
if(q->length){
prev += exec_task(q,quantum);

if(q->length){
q->prev_end = prev;
}

}
}

task = task_save;

while(task){
task = dequeue(task,&q);
// 終了していないプロセスがあるかどうかを調べる

if(q->length > 0){
task = task_save;
r_flag = 1;
break;
}
}
}
}

```

・実行結果

```

[j05002@os-queue]% ./os-queue < data/task_data.03
0 a: 80
80 c: 60
140 b: 80
220 e: 50
270 d: 10
280 a: 80
360 end:

```

```

average waiting time = 238
average turn around time = 254

```

quantumの値を80として実行した。
aのサイズは160だがquantumが80なので80毎に次の処理へ移っていることがわかる。