東南大學
SOUTHEAST UNIVERSITY

OPERATING SYSTEM CONCEPTS

Exercises

A/Prof. Kai Dong

Contents

Q

```c
int value = 5;
int main(int argc, char *argv[]) {
        pid_t pid;
        pid = fork();
        if (pid == 0) {
                printf("child process, value1 : %d\n", value);
                value += 15;
                printf("child process, value2 : %d\n", value);
        }
        else if (pid > 0) {
                printf("parent process, value3 : %d\n", value);
                wait(NULL);
                printf("parent process, value4 : %d\n", value);
        }
        exit(0);
}
```

- What is the output?

A

```
child process, value1: 5
child process, value2: 20
parent process, value3: 5
parent process, value4: 5

parent process, value3: 5
child process, value1: 5
child process, value2: 20
parent process, value4: 5
```

Q

- Including the initial parent process, how many processes are created by the following program.

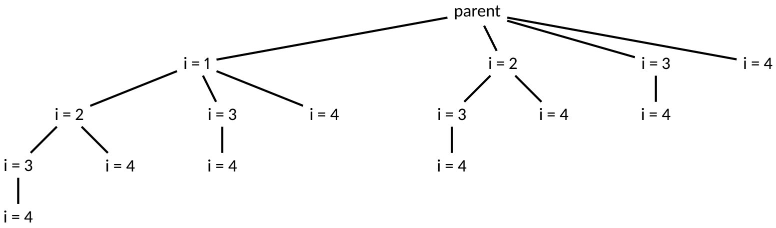
```c
#include <stdio.h>
#include <unistd.h>

int main() {
        int i;
        for (i = 0; i < 4; i++)
                fork();
        return 0;
}
```

A

```c
int i;
for (i = 0; i < 4; i++)
  fork();
return 0;
```

```
0x1024 ...; some instructions
         implementing fork()
0x2028 incl %eax
0x202c cmpl $0x0004, %eax
0x2030 jne 0x1024
0x2032 retn
```



Q

- Draw the diagram of process state.

Contents

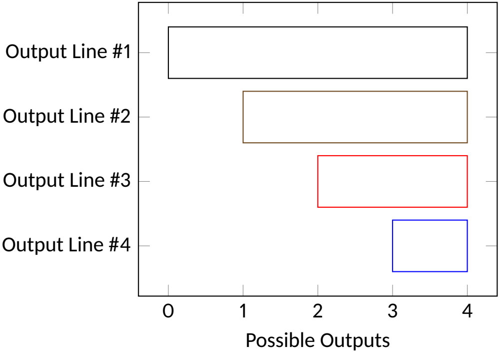
In Class Exercise

```c
/* kai.c */
#include <stdio.h>
#include <pthread.h>
void *helloFunc(void *ptr) {
        int *data;
        data = (int *) ptr;
        printf("I'm Thread %d\n", *data);
        return (void *) data;
}
int main(int argc, char *argv[]) {
        pthread_t hThread[4];
        int *rvals[4];
        for (int i = 0; i < 4; i ++)
                pthread_create(&hThread[i], NULL, helloFunc, (void *) &i);
        for (int i = 0; i < 4; i ++) {
                pthread_join(hThread[i], (void **) &rvals[i]);
                //printf("Thread %d returns %d\n", i, *rvals[i])
        }
        return 0;
}
```

```
prompt> gcc -o kai kai.c -pthread -Wall
prompt> ./kai
```

A

Any possible order of a non-decreasing sequence satisfying . . .



Contents

Q

Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

The processes are assumed to have arrived in the order of P_1, P_2, P_3, P_4, P_5, all at time 0.

1. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a larger priority number implies a higher priority), and RR (quantum $= 2$).
2. What is the average turnaround time for each of the scheduling algorithms in part 1?
3. What is the waiting time of each process for each of these scheduling algorithms?
4. Which of the algorithms results in the minimum average waiting time (over all processes)?

A

Q1:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
FCFS	P_1		P_2	P_3								P_4				P_5				
SJF	P_2	P_1		P_4				P_5					P_3							
PRIO	P_3								P_5					P_1		P_4				P_2
RR	P_1		P_2	P_3		P_4		P_5		P_3		P_4		P_5		P_3		P_5	P_3	

Q2:

	FCFS	SJF	PRIO	RR
$T_{turnaround}$ (ms)	$\frac{2+3+11+15+20}{5}$ =10.2	$\frac{1+3+7+12+20}{5}$ =8.6	$\frac{8+13+15+19+20}{5}$ =15	$\frac{2+3+20+13+18}{5}$ =11.2

Q3 & Q4:

		P_1	P_2	P_3	P_4	P_5	Avg	
$T_{waiting}$ (ms)	FCFS	0	2	3	11	15	6.2	
	SJF	1	0	12	3	7	4.8	Minimum
	PRIO	13	19	0	15	8	11	
	RR	0	2	12	9	13	7.2	

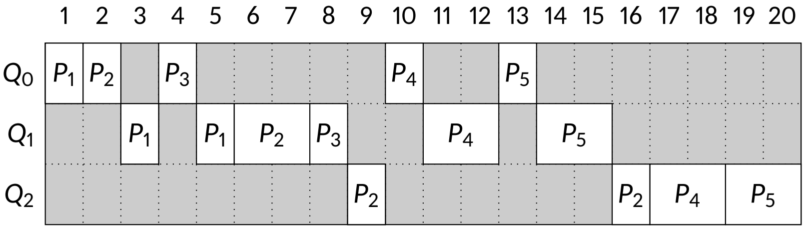
Contents

Q

Given the table below showing the process information of a system, based on multilevel feedback queuing scheduling scheme. Suppose there are five levels in the system and within each level the FCFS scheduling is used. If a process is preempted, it will wait for being scheduled at the first place in the FCFS queue. Draw a Gantt chart to show the time of CPU allocated to each process until all processes are finished using time quantum $q = 2^i$, (where q = time allocated for a process to run in its turn, and i ranges from 0 to 4 indicating the i^{th} level queue).

Process	Arrival Time	Service Time
P_1	0	3
P_2	1	5
P_3	3	2
P_4	9	5
P_5	12	5

A


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Q0
Q1
Q2
P1 P2 P3 P4 P5
P1 P1 P2 P3 P4 P5
P2 P2 P4 P5

Contents

Q

A computer system has 3 types of resources A, B, and C with different numbers of instances. There are 4 running processes P_1, P_2, P_3, P_4. The total resources, the resource's *Allocation* and *Max* matrices for the four processes are shown as follows:

	Allocation			*Max*		
Process	*A*	*B*	*C*	*A*	*B*	*C*
P_1	1	3	1	6	5	3
P_2	0	2	2	3	5	3
P_3	2	0	0	3	5	2
P_4	0	1	3	2	4	3
total	6	9	6			

1. What are the matrices *Need* and *Available* for the system?
2. Please check if the system is currently deadlocked. Show your steps clearly.
3. At the current state, if P_2 requests additional resources $[1, 0, 0]$, can the request be granted without any possible deadlock? Show your steps clearly.
4. At the current state, if P_1 requests additional resources $[1, 0, 0]$, can the request be granted without any possible deadlock? Show your steps clearly.

A

- $n = 4$, $m = 3$, *Allocation* and *Max* are already defined.
- Key to **Q1**: Compute *Available* and *Need*

Process	*Need* A	B	C
P_1	5	2	2
P_2	3	3	1
P_3	1	5	2
P_4	2	3	0
Available	3	3	0

- Key to **Q3 Q4**: Resource-Request Alg.
- Assume the request is granted, use the Safety Alg. to determine whether the system is in safe state.

- Key to **Q2**: Safety Alg.
  1. $Work = Available$, $Finish[i] = false, 0 \leq i < n$.
  2. Find an $i = 4$.
  3. $Work = Work + Allocation_i$, $Finish[4] = true$

	A	B	C
Work	3	4	3

  4. Go to step 2.
  5. Find an $i = 2$.
  6. $Work =< 3, 6, 5 >$, $Finish[2] = true$
  7. Go to step 2.
  8. Find an $i = 3$.
- Safe sequence: $< P_4, P_2, P_3, P_1 >$

Q

Consider the following system snapshot using the data structures in the Banker's algorithm, with resources A, B, C, and D, and processes P0 to P4:

	Max				Allocation				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P0	2	3	3	3	1	2	1	2								
P1	1	4	1	0	1	1	0	0								
P2	2	1	1	1	0	1	0	1								
P3	5	4	3	3	1	1	2	2								
P4	4	2	6	3	1	2	1	2								
Total Res													2	0	2	0

1. How many resources of type A, B, C, and D are there?
2. What are the contents of the Need matrix?
3. Is the system in a safe state? Why?
4. If a request from process P2 arrives for additional resources of (0,0,2,0), can the Banker's algorithm grant the request immediately? Why?

Contents

Q

Consider a system with 64 MB of physical memory, 32-bit physical addresses, 32-bit virtual addresses, and 4 KB physical page frames.

(a) Using a single-level paging scheme, what is the maximum number of page table entries for a page table in this system?

(b) Using a two-level paging scheme with a 1024-entry outer-page table, how many bits are needed in the logical address to represent the outer page table? How many bits are needed in the logical address in order to represent the inner page table? How many bits are used to represent the offset within a page?

(c) Suppose a TLB is used with the two-level paging scheme described in part (b), and the TLB has a 90% hit rate. If the TLB access time is 10 ns and memory access time is 100 ns, what is the effective memory access time of the system?

A

Consider a system with 64 MB of physical memory, 32-bit physical addresses, 32-bit virtual addresses, and 4 KB physical page frames.

(a) Using a single-level paging scheme, what is the maximum number of page table entries for a page table in this system?

2^{20}

(b) Using a two-level paging scheme with a 1024-entry outer-page table, how many bits are needed in the logical address to represent the outer page table? How many bits are needed in the logical address in order to represent the inner page table? How many bits are used to represent the offset within a page?

10, 10, 12

(c) Suppose a TLB is used with the two-level paging scheme described in part (b), and the TLB has a 90% hit rate. If the TLB access time is 10 ns and memory access time is 100 ns, what is the effective memory access time of the system?

$EAT = 90\% \times (10 + 100) + 10\% \times (20 + 300) = 99 + 32 = 131$

Contents

Q

Consider the reference page sequence is 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5, and the number of page frame is 3.

(a) How many page faults for FIFO algorithm?

(b) How many page faults for LRU algorithm?

(c) How many page faults for OPT algorithm?

A

Consider the reference page sequence is 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5, and the number of page frame is 3.

(a) How many page faults for FIFO algorithm? Key: 9

	1	2	3	4	1	2	5	1	2	3	4	5
	1	1	1	4	4	4	5	5	5	5	5	5
		2	2	2	1	1	1	1	1	3	3	3
			3	3	3	2	2	2	2	2	4	4
PF	∘	∘	∘	∘	∘	∘	∘			∘	∘	

(b) How many page faults for LRU algorithm? Key: 10

	1	2	3	4	1	2	5	1	2	3	4	5
	1	1	1	4	4	4	5	5	5	3	3	3
		2	2	2	1	1	1	1	1	1	4	4
			3	3	3	2	2	2	2	2	2	5
PF	∘	∘	∘	∘	∘	∘	∘			∘	∘	∘

(c) How many page faults for OPT algorithm? Key: 7

	1	2	3	4	1	2	5	1	2	3	4	5
	1	1	1	1	1	1	1	1	1	3	4	4
		2	2	2	2	2	2	2	2	2	2	2
			3	4	4	4	5	5	5	5	5	5
PF	∘	∘	∘	∘			∘			∘	∘	

Contents

Q

Suppose that a disk drive has 200 cylinders, numbered 0 to 199. The drive is currently serving a request at cylinder 123, and the previous request was at cylinder 175. The queue of pending requests, in FIFO order, is 86, 147, 180, 28, 95, 151, 12, 77, 30. Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms respectively?

1. FCFS
2. SSTF
3. C-SCAN
4. LOOK