Operating systems
Final examination
(part 2)
2009/2010

Student Name:

NIA:

1. (2pt) Suppose there are 3 robots (red, blue, green), each of which is controlled by its own thread. You must ensure that the robots only move in the following order: red, blue, green, red, blue, green, etc. Write the pseudo-code that performs the appropriate initializations and enforces this execution order. Use only semaphores for your synchronization.

Answer:

```
Semaphore r=1, b=0, g=0
red_robot{
  while (1) {
    wait(r);
    move;
    signal(b);
  }
}
blue_robot{
  while (1) {
    wait(b);
    move;
    signal(g);
  }
}
green_robot{
  while (1) {
    wait(g);
    move;
    signal(r);
  }
```

2. (2 pt) Is the following system of four processes with 2 resources deadlocked? Justify your answer (no justification receives 0 credit).

Current allocation matrix

P1 1 3

P2 4 1
P3 1 2
P4 2 0

Current request matrix
P1 1 2
P2 4 3
P3 1 7
P4 5 1

Availability Vector
1 4

Answer:

Give 1 2 to P1:
Current allocation matrix
P1 2 5
P2 4 1
P3 1 2
P4 2 0
Current request matrix
P1 0 0
P2 4 3
P3 1 7
P4 5 1
Availability
0 2

P1 finishes, releases 2 5.
Current allocation matrix
P2 4 1
P3 1 2
P4 2 0
Current request matrix
P2 4 3
P3 1 7
P4 5 1
Availability
2 7

Give 1 7 to P3:
Current allocation matrix
P2 4 1
P3 2 9
P4 2 0
Current request matrix

P2 4 3
P3 0 0
P4 5 1
Availability
1 0

P3 finishes, releases 2 9:
Current allocation matrix
P2 4 1
P4 2 0
Current request matrix
P2 4 3
P4 5 1
Availability
3 9

Now there is no possible request that can be fulfilled, so it is deadlocked.

3. (2pt) Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?

Answer:

a. First-fit:

212K is put in 500K partition

417K is put in 600K partition

112K is put in 288K partition (new partition 288K = 500K − 212K)

426K must wait

b. Best-fit:

212K is put in 300K partition

417K is put in 500K partition

112K is put in 200K partition

426K is put in 600K partition

k. Worst-fit:

212K is put in 600K partition

417K is put in 500K partition

112K is put in 388K partition

426K must wait

In this example, best-fit turns out to be the best.

4. (2pts) Suppose that you have a UNIX file system where the disk block size is 1kB, and an inode takes 128 bytes. Disk addresses take 32 bits, and the inode contains space for 64 bytes of data, 8 direct addresses, one indirect, one double-indirect and one triple- indirect (the rest of the space in the inode is taken up with other information such as ownership and protection). An index block is the same size as a disk block. How much space (including overhead) do files that are: a) one (1) byte long, b) 1025 bytes long, c) 65536 (64KB) bytes long, and d) 1048576 (1MB) bytes long require?
Hint: it may help if you draw a picture of how inodes are used to locate the blocks making up a file.

Answer:

Disk block: 1kB = 1024 bytes
Inode: 128 bytes (64 bytes of data, 64 bytes of metadata)
Address: 32 bits (4 bytes)
Index block: same size as a disk block, 1024 bytes

1 byte long file: 1 inode (the data fits in the 64 bytes in the inode)
1*128
= 128 bytes
= 1 byte of data + 127 bytes of overhead

1025 byte long file: 1 inode + 1 disk block (pointed to by the inode)
128 + 1024 bytes
= 1152 bytes
= 1025 bytes of data and 127 bytes of overhead

65536 (64KB) bytes long file: 1 inode + 8 data blocks + 1 index block (pointed to by the indirect pointer) + 56 data blocks
128 + 8*1024 + 1024 + 56*1024
= 66688 bytes
= 64KB of data and 1152 bytes of overhead

1048576 (1MB) bytes long file: 1 inode + 8 data blocks + 1 index block (pointed to by the indirect pointer) + 256 data blocks + 1 index block (pointed to by the double indirect pointer) + 3 index blocks (pointed to by the double indirect index) + 256 blocks (pointed to by the first of the three index blocks)+ 256 blocks (pointed to by the second of the three index blocks)+ 248 blocks (pointed to by the third of the three index blocks)
128 + 8*1024 + 1024 + 256*1024 + 1024 + 3*1024 + 256*1024 + 256*1024 + 248*1024
= 1053824 bytes
= 1MB of data and 5248 bytes of overhead

5. (1pt) Define external and internal fragmentation (with respect to file systems).

Answer:

External fragmentation: occurs under file systems in which files are allocated as contiguous sets of blocks: small, unused holes accumulate within the file system that are not large enough to fit long files (even though collectively there may be enough free space).

Internal fragmentation: space within a block that is unused for the file data.

6. (1pt) Define the four necessary conditions for a deadlock to arise.

Answer:

Mutual exclusion

Hold and wait

No preemption

Circular Wait