Homework 4 Solution

cs550 – Operating Systems

1.

a.

0 30 60 90 120 150 180 210 240 270 300 330 360 390 420 450 464 468 564

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

| P1| P2| P3| P4| P5| P1| P3| P4| P5| P1| P3| P4| P5| P1| P4| P5| P1| P4|

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

b.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 36 | 134 | 261 | 401 614 |
| P2 | P3 | P1 | P5 | P4 |

c.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 28 | 117 | 204 | 312 | 432 609 |
| P2 | P3 | P6 | P5 | P4 | P1 |

2.

a.

Total resources – (2, 3, 2, 2)

Current allocations:

P1 (0, 1, 0, 1)

P2 (1, 1, 0, 0)

P3 (0, 0, 1, 0)

Total used resources (1, 2, 1, 1)

Total free resources (1, 1, 1, 1)

Maximum claims:

P1 (1, 1, 0, 1)

P2 (1, 2, 0, 0)

P3 (1, 2, 2, 1)

Give P1 (1, 0, 0, 0). P1 has its maximum claim and is assumed to finish.

P1 frees the resources it has acquired.

Now the total free resources are (1, 2, 1, 2).

Give P2 (0, 1, 0, 0). P2 has its maximum claim and is assumed to finish.

P2 frees the resources it has acquired.

Now the total free resources are (2, 3, 1, 2).

Give P3 (1, 2, 1, 1). P3 has its maximum claim and is assumed to finish.

P3 frees the resources it has acquired.

Now all resources are free. All processes are able to complete – safe state.

b.

Total resources – (4, 4, 2)

Current allocations

P1 (0, 0, 0)

P2 (1, 1, 0)

P3 (2, 0, 0)

Maximum claims

P1 (2, 0, 1)

P2 (3, 2, 0)

P3 (3, 0, 1)

Used resources (3, 1, 0)

Available resources (1, 3, 2)

Give P3 (1, 0, 1) and P3 acquires its maximum claim.

P3 is assumed to finish and frees its resources.

Now (3, 3, 2) is free.

Give P1 (2, 0, 1) and P1 acquires its maximum claim.

P1 is assumed to finish and frees its resources.

Now (3, 3, 2) is free.

Give P2 (2, 1, 0) and P2 acquires its maximum claim.

P2 is assumed to finish and frees its resources.

Now (4, 4, 2) is free.

Since all processes have finished and freed their resources, this state is assumed to be safe.

c.

Total resources (5, 6, 7, 2)

Current allocations

P1 (1, 2, 0, 1)

P2 (1, 1, 4, 0)

P3 (1, 1, 0, 0)

P4 (0, 0, 2, 0)

Maximum claims

P1 (2, 2, 2, 2)

P2 (2, 3, 4, 1)

P3 (1, 3, 0, 0)

P4 (1, 3, 2, 1)

Total used – (3, 4, 6, 1)

Total free – (2, 2, 1, 1)

P2 can acquire (1, 2, 0, 1) and meet its maximum claim.

Since P2 has acquired its maximum claim, it is assumed to finish and frees its resources. Now, (3, 3, 5, 1) is free.

P1 can acquire (1, 0, 2, 1) and meet its maximum claim.

Since P1 has acquired its maximum claim, it is assumed to finish and frees its resources. Now (4, 5, 5, 2) is free.

P3 can acquire (0, 2, 0, 0) and meet its maximum claim.

Since P3 has acquired its maximum claim, it is assumed to finish and frees its resources. Now (5, 6, 5, 2) is free.

P4 can acquire (1, 3, 0, 1) and meet its maximum claim.

Since P3 has acquired its maximum claim, it is assumed to finish and frees its resources. Now (5, 6, 7, 2) is free.

Since all processes have finished and freed their resources, this state is assumed to be safe.

3. Many answers are valid for this problem. The following is only an example.

Name – ZFS/OpenZFS – Zetabyte File System or Z File System a 128 bit file system

Support for journaling/transaction logging – yes

Maximum volume size – 256 Zettabytes (278 bytes maximum pool size)
Maximum file size – 16Exabytes (264 bytes)
Description of file permissions – This filesystem contains POSIX compliant permissions for user, group, and world access
Discussion of directory structure – Consists of a “uberblock” which is a tree with block pointers and data is only stored in leaves of the tree.
Description of why this file system is useful – This file system is a both a file system and volume manager, meaning that it is possible to represent files spanning across multiple physical disks within a single file system.

Resource - <http://docs.oracle.com/cd/E19253-01/819-5461/zfsover-2/index.html>

<http://www.oracle.com/us/products/servers-storage/solaris/solaris-zfs-ds-067320.pdf>

<http://www.osdevcon.org/2009/slides/zfs_internals_uli_graef.pdf>