

2016 Operating System Examination Solution

1. a. i) Shortest Job First

ii) 10-200 ms.

iii) When the quantum is significantly smaller than the average job length.

iv) When the length of jobs are equal or less than 1 second

b. i) There are 2520 different possible interleavings. (Formula used here: when there are n threads each with m instructions, then there are $\frac{(nm)!}{(m!)^n}$ different interleavings)

ii) The semaphore here is used to synchronize the threads

```
// in T1
int main() {
    down(s);
    down(s);
    down(s);
    a = 1;
    b = 1;
}

// in T2, T3, T4
int main() {
    a = i;
    a = i;
    up(s);
}
```

The `down` and `up` instruction is added as shown above. We add three `down`s in front of the instructions of T1, and add `up` at the end of T2-4. In this way, once T2-4 have completed their execution, T1 can then be executed since it will be blocked by three `down`s before T2-4 finish.

c. (Not covered in second year material in 2020-2021)

2. a. For a file system to be mounted under Linux, it means that another file system is mounted at the indicated mount point, where the folder content of the mount point is the root directory of the new file system. Meanwhile, the mount table is also updated to record metadata such as the file system type of the new FS.

b. The inode of that file, the number of hard links, the owner of the file, file size, and time of creation, time of last access, time of last modification, etc.

c. The system will first be trapped into the kernel mode to execute the `open` system call. Then the system call will determine the nature of this file: its location and which file system it belongs to. Once the kernel realises that the file is in a mounted file system, it will go through the information in the mount table to find out what type of file system it uses. Next, it will go through the directory tree of the new file system and find the inode of the file. Finally, we add the inode, or the file representation, to a list of files opened by the current thread and then return its appropriate file descriptor.

d. `wombat.txt` could only be a soft link to `platypus.txt` since `wombat.txt` lives in a different file system than `platypus.txt`. They might have different inode representations or file system organizations, hence hard link is not suitable here.

e. In order to store hard links across different file systems, the Linux kernel has to develop a strategy to unify the inode representation across different file systems (make them homogeneous). We might need a wrapper data structure with basic inode information in it, as well as the type of file system, mount point, etc. When mounted under `/mnt`, the homogeneous inode representations of files under `/mnt` have to be generated in order to create hard links to original file system.