EURECOM
Sophia Antipolis

Exam
Operating Systems - OS

Ludovic Apvrille
ludovic.apvrille@telecom-paristech.fr

February, 2016

Authorized documents: Nothing! The grading takes into account the fact that you don't have any documents with you.
A grade is provided for every question (beware: do organize your time, e.g., last question is a 4-point question). 1 additional point is given as a general appreciation, including written skills and readability.

1 Course understanding (7 points, ∼30 minutes)

a. What is the interest of double buffering for drivers? Provide an example to illustrate your answer. [3 points]

b. A process may contain several threads. What are the memory areas which are shared by the threads of one given process, and which areas belong only to one thread? Why is it so? Provide a pseudo-code that illustrates the sharing of a memory area - e.g. a variable - between two processes, and between two threads of the same process. [4 points]

2 Linux Ram disk (11 points, ∼90 minutes)

File partitions in Linux can also be set directly in RAM. Below, you will find an article discussing two different RAM-based file systems (ramfs, tmpfs). Carefully read this article, and answer to the following questions.

a. Provide a summary of this article in 150 words (+/- 10 words). [4 points]

b. Comment on the current memory/swap state of the Linux machine of the author. [2 points]

c. Could you be more precise on what the author meant by "As the data is lost when the machine reboots the data must not be precious as even scheduling backups cannot guarantee that all the data will be replicated in the event of a system crash" [2 points]

d. I have done the following test on my Linux machine: I have compiled the latex sources of the RTOS slides from the same state (i.e., after removing all intermediate files of the compilation process) in two cases: (i) with the files located on my SSD, and (ii) with the same files located in a Ramdisk. **Comment on the results** with regards to what is stated in the article. **Imagine a better test** to evaluate the difference between an SSD and a ramfs. [4 points]

SSD:

```
$ make ultraclean&&time make all
real    0m34.464s
user    0m22.256s
sys     0m6.528s
```

Ramdisk (tmpfs): I have first created the ramdisk, then, I have mounted it in /mnt/ramdisk. Finally, I have copied the files from the SSD and compiled the latex sources:

```
$ sudo mkdir /mnt/ramdisk
$ sudo mount -t tmpfs -o size=1024m tmpfs /mnt/ramdisk
$ cd /mnt/ramdisk
$ cp -R /homes/apvrille/slidesRTOS .
$ make ultraclean&&time make all
real    0m26.788s
user    0m21.192s
sys     0m6.052s
```

The Difference Between a tmpfs and ramfs RAM Disk

Dec 2013.

Taken from http://www.jamescoyle.net/knowledge/951-the-difference-between-a-tmpfs-and-ramfs-ram-disk

There are two file system types built into most modern Linux distributions which allow you to create a RAM based storage area which can be mounted and used like a normal folder.

Before using this type of file system you must understand the benefits and problems of memory file system in general, as well as the two different types. The two types of RAM disk file systems are tmpfs and ramfs and each type has it's own strengths and weaknesses.

What is a memory based file system (RAM disk)?

A memory based file system is something which creates a storage area directly in a computers RAM as if it were a partition on a disk drive. As RAM is a volatile type of

memory which means when the system is restarted or crashes the file system is lost along with all it's data.

The major benefit to memory based file systems is that they are very fast – 10s of times faster than modern SSDs. Read and write performance is massively increased for all workload types. These types of fast storage areas are ideally suited for applications which need repetitively small data areas for caching or using as temporary space. As the data is lost when the machine reboots the data must not be precious as even scheduling backups cannot guarantee that all the data will be replicated in the event of a system crash.

tmpfs vs. ramfs

The two main RAM based file system types in Linux are tmpfs and ramfs. ramfs is the older file system type and is largely replaced in most scenarios by tmpfs.

ramfs creates an in memory file system which uses the same mechanism and storage space as Linux file system cache. Running the command free in Linux will show you the amount of RAM you have on your system, including the amount of file system cache in use. The below is an example of a 31GB of ram in a production server.

```
$ free -g
                    total  used  free  shared  buffers  cached
Mem:                   31    29     2       0        0       8
-/+ buffers/cache:            20    11
Swap:                  13     6     7
```

Note: free displays the total amount of free and used physical and swap memory in the system, as well as the buffers used by the kernel. The "-g" option is used to display the amount of memory in gigabytes.

Currently 8GB of file system cache is in use on the system. This memory is generally used by Linux to cache recently accessed files so that the next time they are requested then can be fetched from RAM very quickly. ramfs uses this same memory and exactly the same mechanism which causes Linux to cache files with the exception that it is not removed when the memory used exceeds threshold set by the system.

ramfs file systems cannot be limited in size like a disk base file system which is limited by its capacity. ramfs will continue using memory storage until the system runs out of RAM and likely crashes or becomes unresponsive. This is a problem if the application writing to the file system cannot be limited in total size. Another issue is you cannot see the size of the file system in df and it can only be estimated by looking at the cached entry in free. tmpfs

tmpfs is a more recent RAM file system which overcomes many of the drawbacks with ramfs. You can specify a size limit in tmpfs which will give a 'disk full' error when the limit is reached. This behaviour is exactly the same as a partition of a physical disk.

The size and used amount of space on a tmpfs partition is also displayed in df. The below example shows an empty 512MB RAM disk.

```
$ df -h /mnt/ramdisk
Filesystem Size    Used   Avail     Use% Mounted on
tmpfs      512M      0     512M      0%   /mnt/ramdisk
```

These two differences between ramfs and tmpfs make tmpfs much more manageable however this is one major drawback; tmpfs may use SWAP space. If your system runs out of physical RAM, files in your tmpfs partitions may be written to disk based SWAP partitions and will have to be read from disk when the file is next accessed. In some environments this can be seen as a benefit as you are less likely to get out of memory exceptions as you could with ramfs because more 'memory' is available to use.