SO EXAM 24 IUN 2019

1. grep -E "[0-24]+(:[0-59]){1,}" a.txt

SO EXAM 27 IUN 2019

1. grep -E "([0-1]){3,}00" binar
2. sed -E "s/([13579])([aeiou])/\2\1/g" a.txt
3. cat scor | sort | uniq -u

SO EXAM EX1:

1. grep -E "^[A-Z]" a.txt
2. sed -E "s/([0-9])([0-9])/\2\1/gi" a.txt
3. awk '{print s=$1 + $2 }' number
4. cat a.txt | sort | uniq -u

files=`find ./ -maxdepth 1 -type f`

ok=0

for file in $files

do

IFS=$'\n'

ok=0

for line in ` cat $file `

do

if echo $line | grep -E -q "cat"

then

ok=1

fi

done

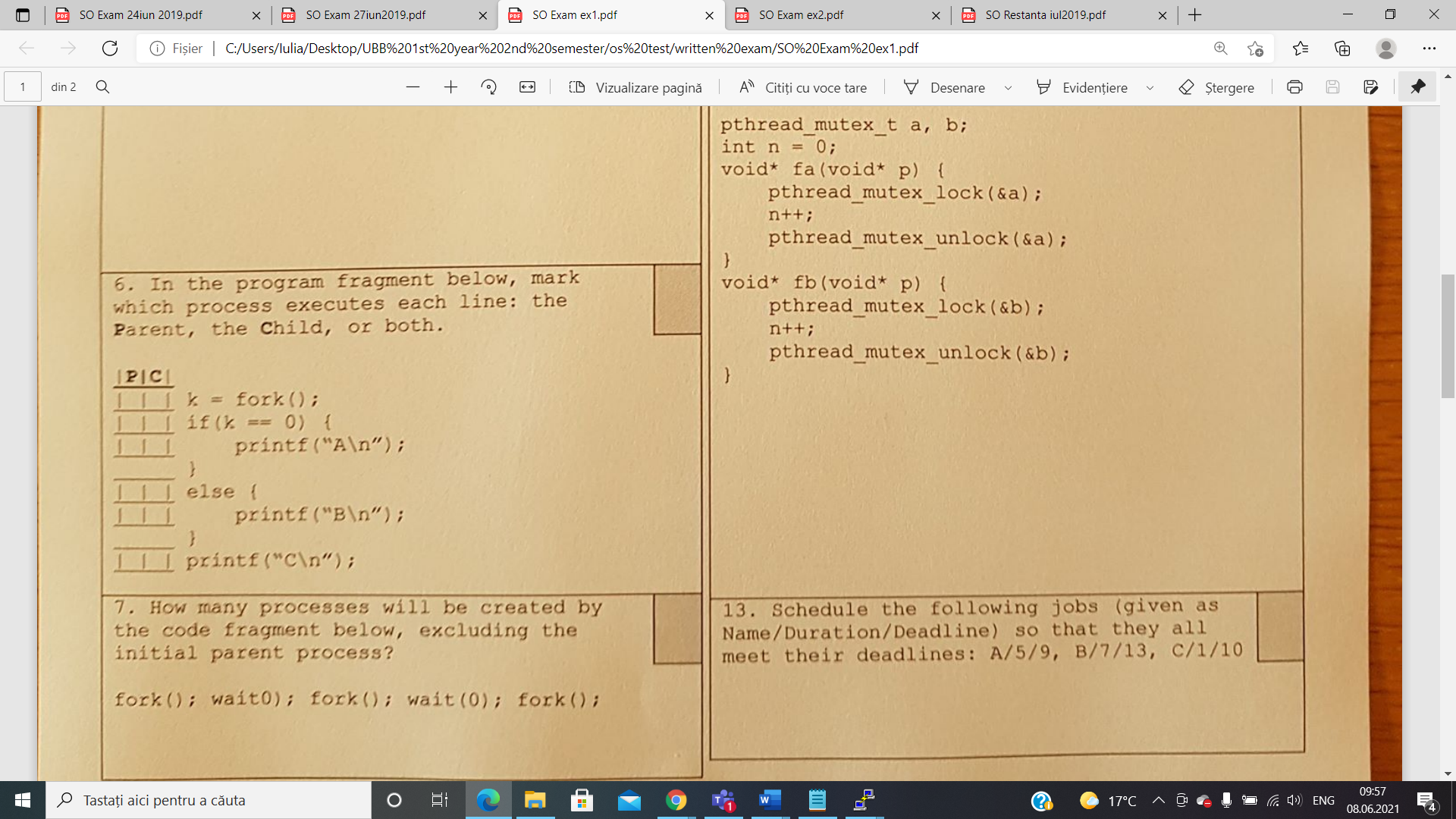
if [ $ok -eq 1 ]

then

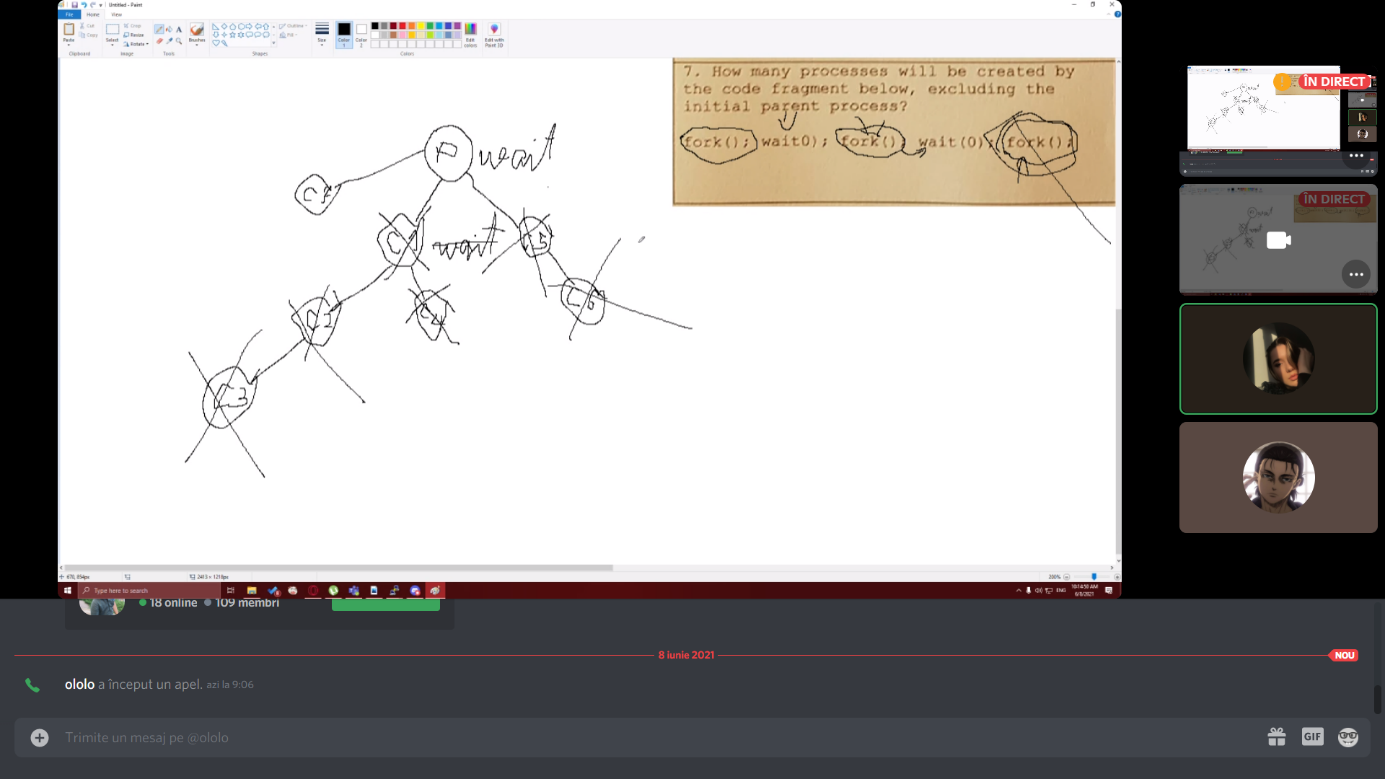
echo $file

fi

done

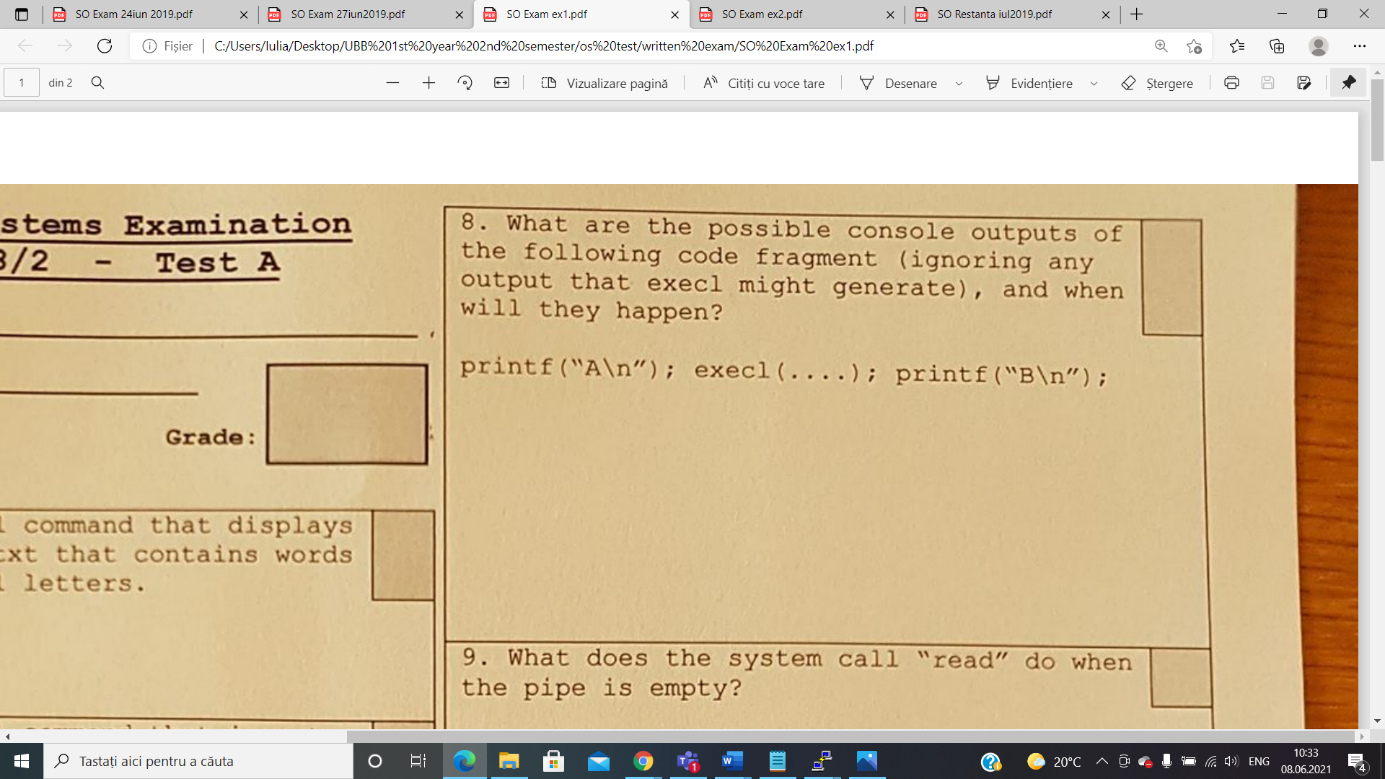




1. 



R: 7.

1. 

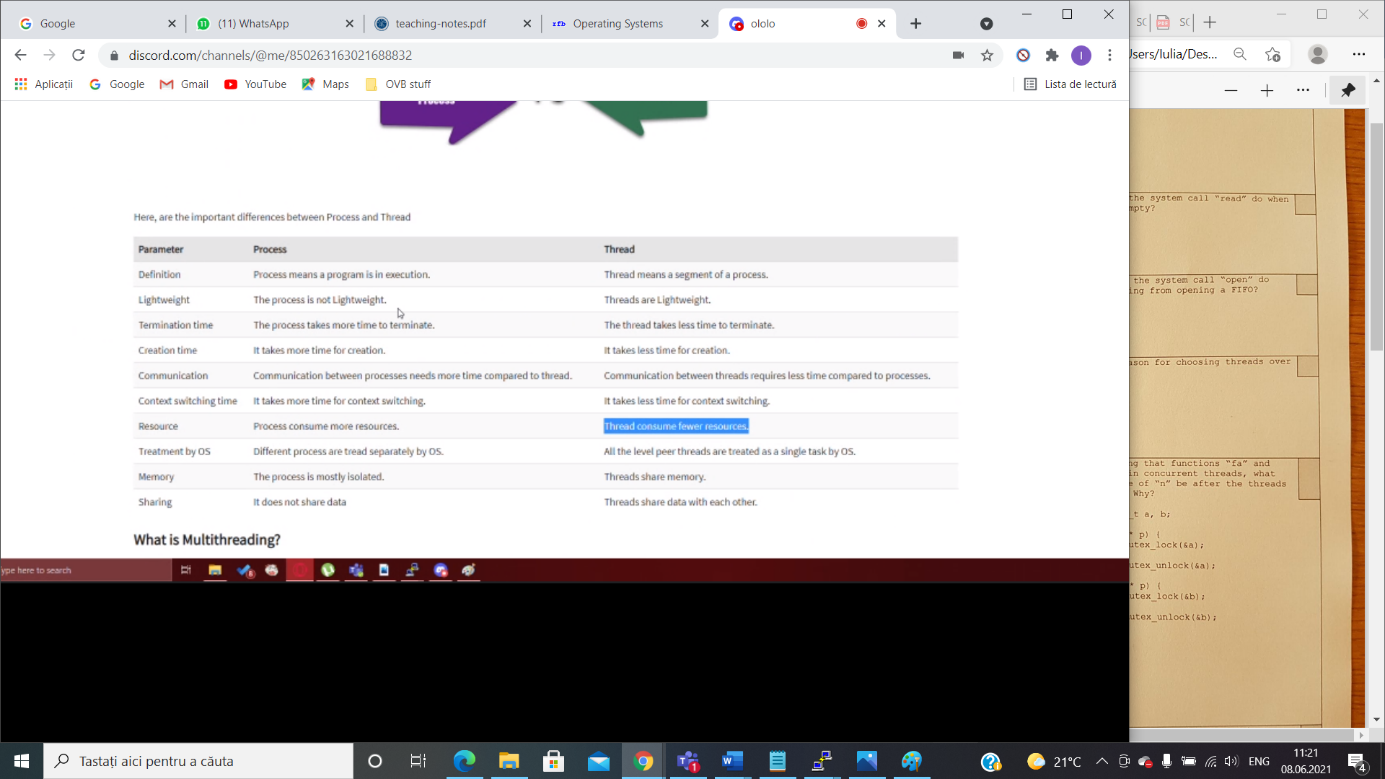
Case 1: (execl succeeds)

* The output will be A because execl commands wipes the initial code and replace it with the code of the new program

Case 2: (execl does not succeeds)

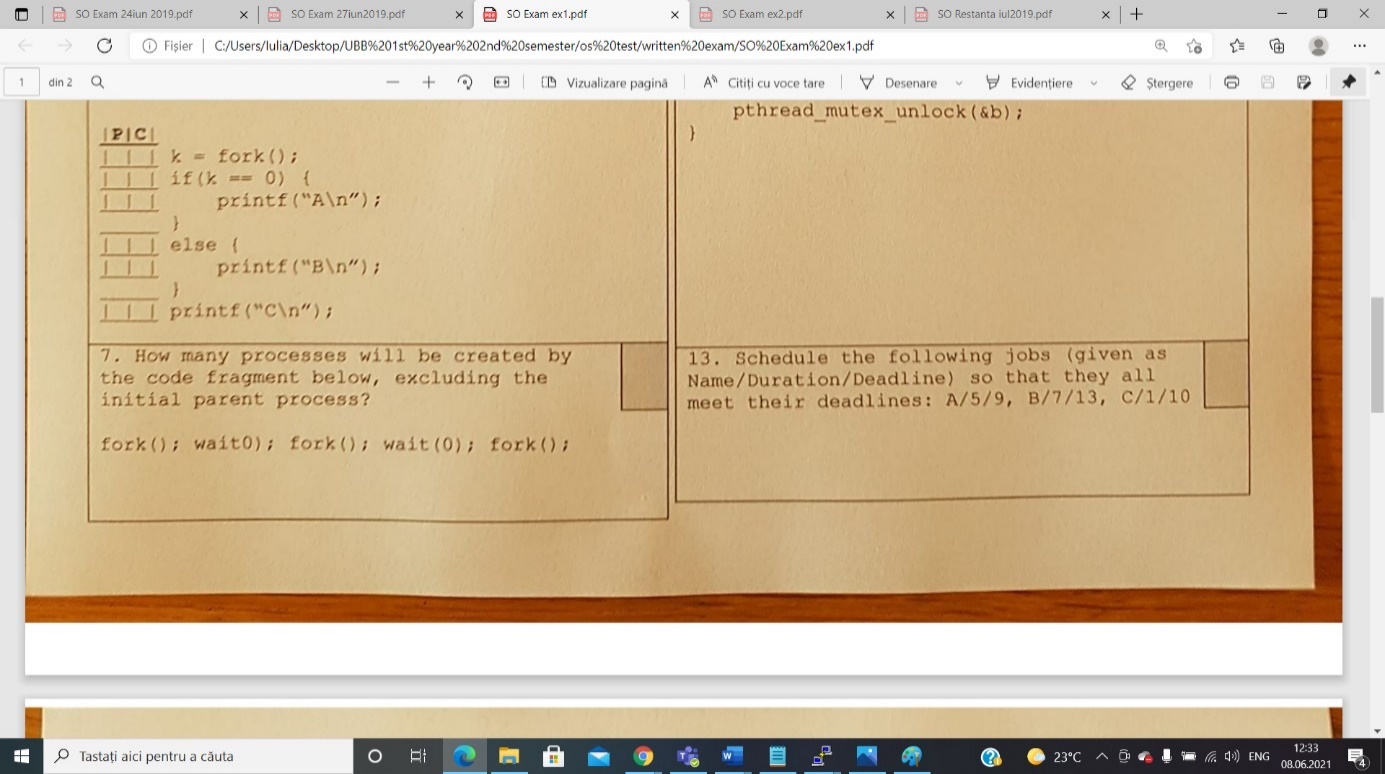
* The output will be A \*newline\* B because the execl call failed so the calling process continues to execute

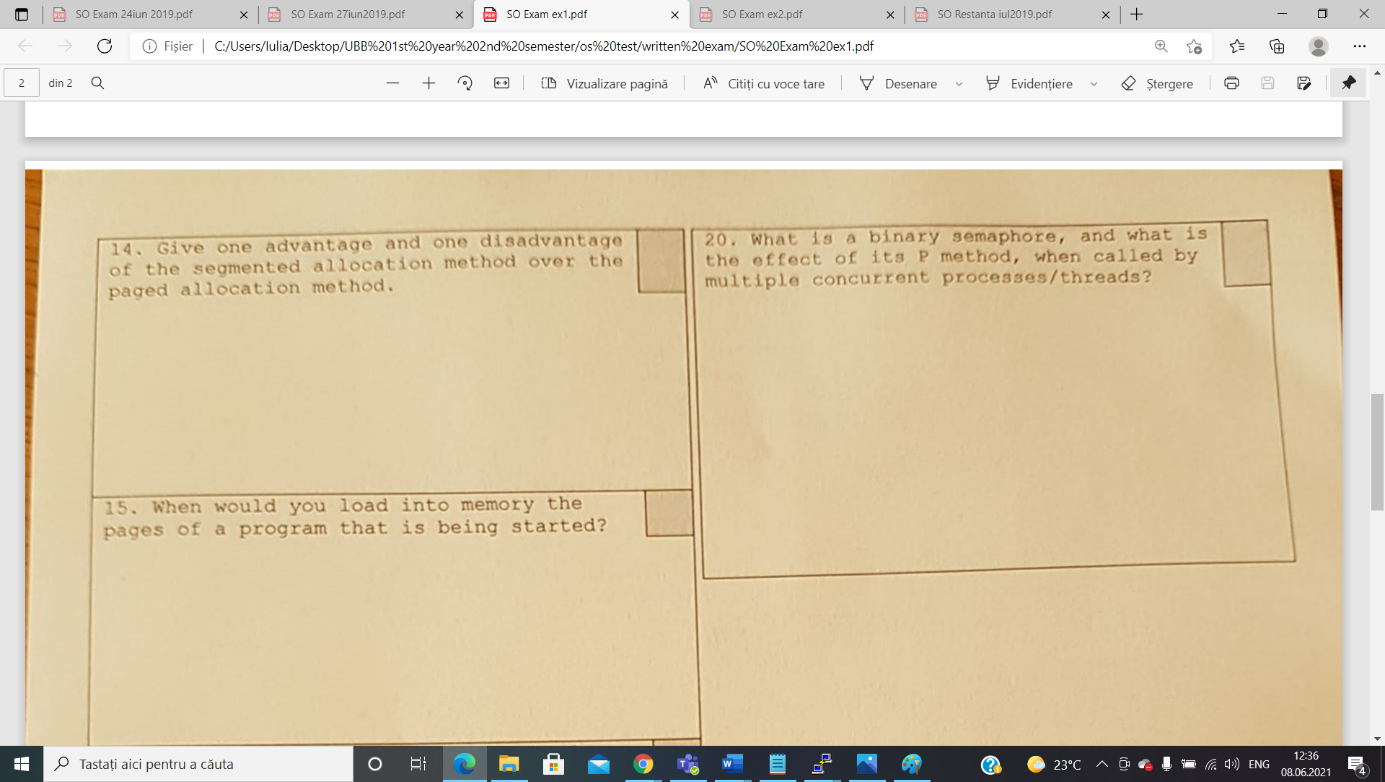
1. The system call will wait for data if the is a writer connected or the pipe will see the end of file (read will return 0)
2. The system call will wait until the complementary operation in the other process is opened.
3. Sometimes it’s better to use threads instead of processes because threads use less resources taking into consideration that they have a shared memory, they use global variables etc, whereas processes does not share data, consume more resources and takes more time for creation and communication between them.



1. The result is uncertain because there is no efficient synchronization between the threads as there are 2 global mutexes which are used in two different functions instead of one used globally in both of them.

A/5/9; C/1/10; B/7/13





Disadvantage:

* The segmented allocation method is slower than the paging one, in terms of memory access

Advantage:

* The segmented allocation method has a better protection of memory because each segment may receive other access rights.

1. A binary semaphore is just like a mutex lock, it has 2 values( 0 and 1) and it is initialised with 1. Concurrent threads will wait at the binary semaphore if its value is 0.  A semaphore is a signaling mechanism, and a thread that is waiting on a semaphore can be signaled by another thread. It uses two atomic operations, 1)wait, and 2) signal for the process synchronization.

The P method makes concurrent threads wait at the semaphore if its value is 0 (P=wait, sleep, down operation)

SO EXAM EX2:

1. sed -E "s/([^A-Z0-9a-z])([A-Z0-9a-z])/\2\/gi" a.txt ***SAU*** sed -E "s/([^A-Z0-9a-z])([A-Z0-9a-z])/\2/gi" a.txt
2. awk 'BEGIN{s=0} {s = s + NF } END{print s/NR}' number
3. find -type f | awk -F / '{print $NF}' | sort | uniq -u

files=`find ./ -type f`

nr=0

s=0

for file in $files

do

if echo $file | grep -E -q ".txt"

then

a=`cat $file | wc -l`

s=`expr $s + $a`

nr=`expr $nr + 1`

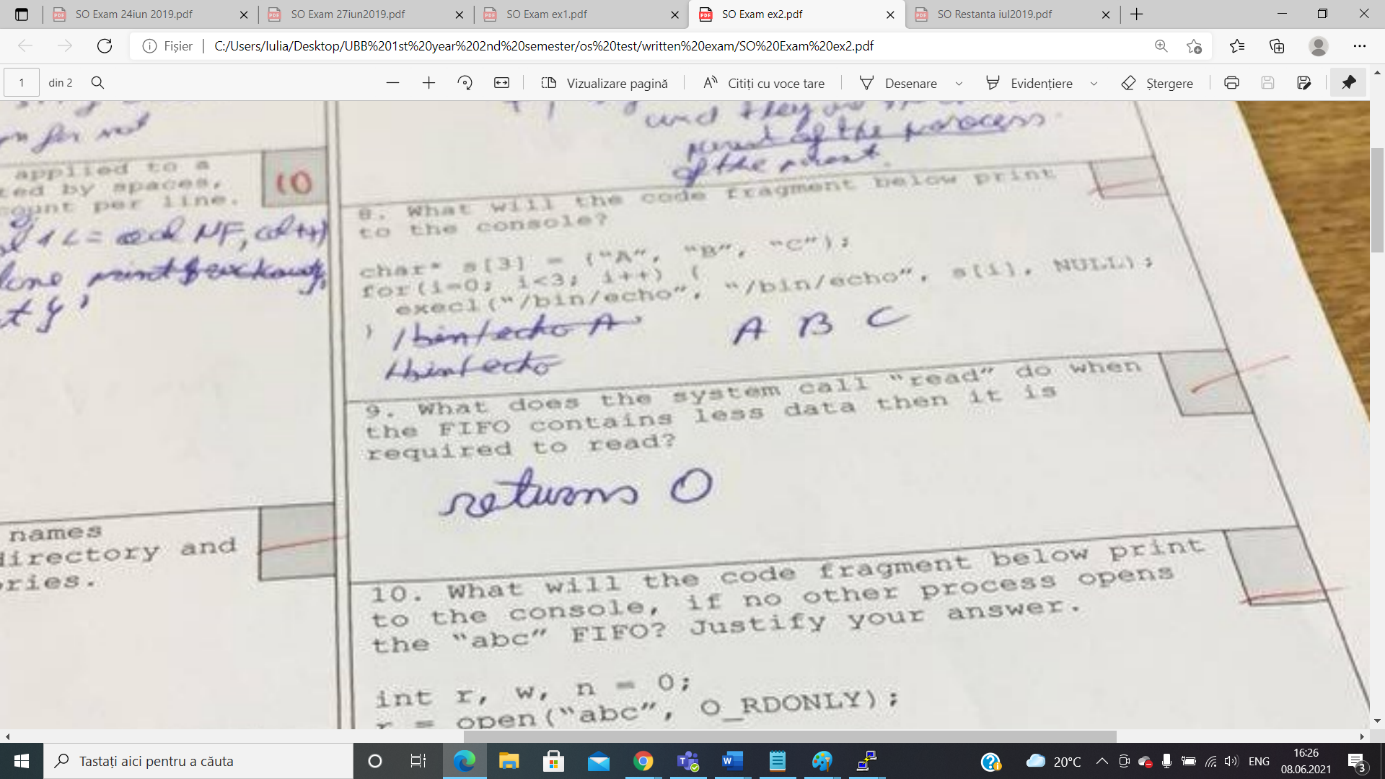
fi

done

s=`expr $s / $nr`

echo $s

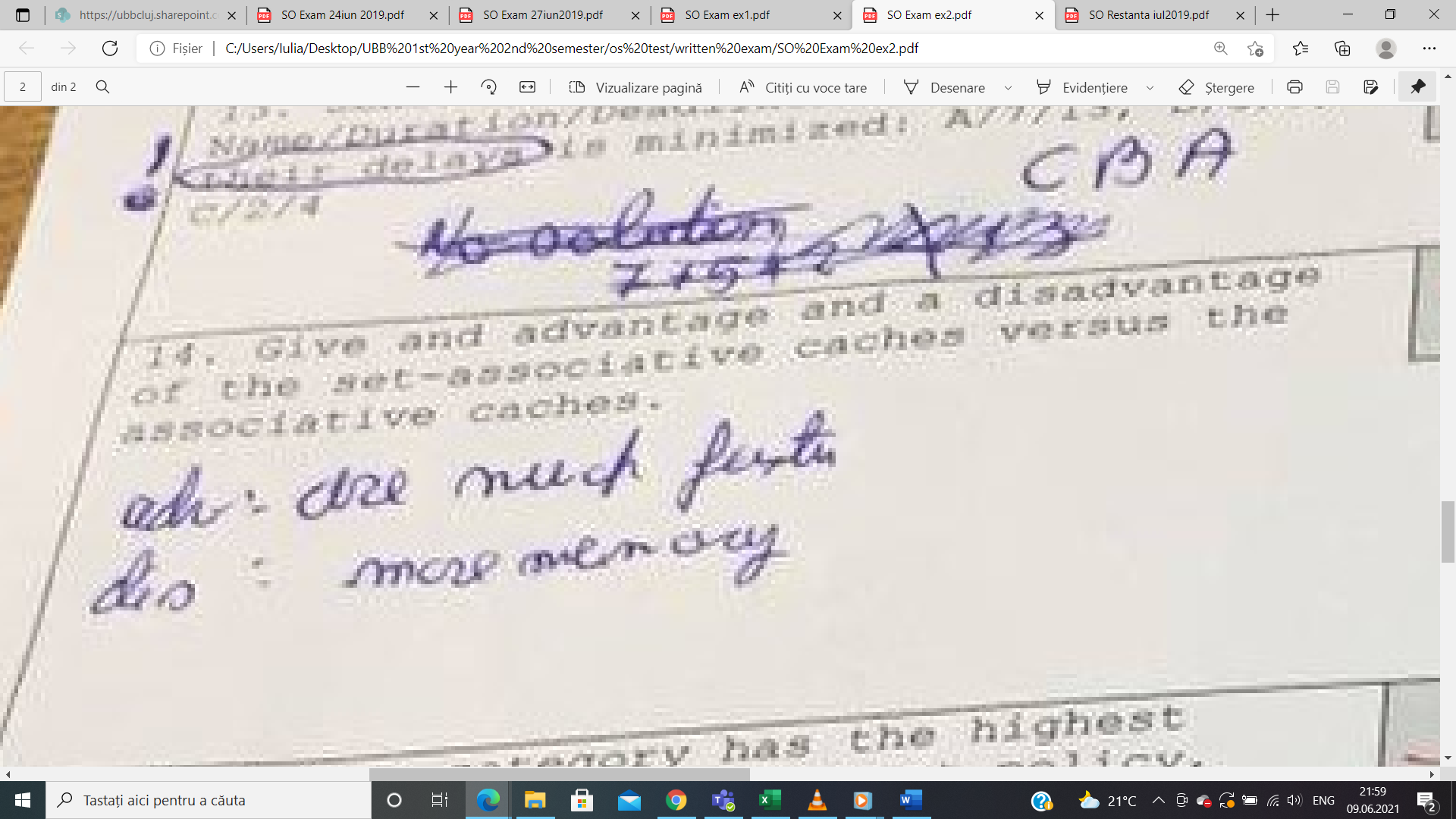
1. A



It will try to read as many bytes as they are at the moment of reading and then will return what it could read.

INVERS: Some bytes will be overwritten and the result will be messed up.

1. The program will be stuck and the program will not return anything.

4. C, B, A
5. 

ADVANTAGE:

DISADVANTAGE:

**SUBIECT 2020**

1. Avantaj si dezavantaj al alocarii cu partitii variablie si alocarea paginata

* **Avantaj**: Considering the fact that the address doesn’t need to be translated from a virtual address to a physical address we can say that partitioned allocation is faster from a computational point of view
* **Dezavantaj**: Apare fenomenul de fragmentare internA a memoriei deoarece procesele aloca memorie iar apoi sunt eliberate, Ramanand spatii libere care nu vor fi ocupate deoarece dimensiune lor va scadea.



The most prioritary class of pages that would be chosen as a victim by the NRU policy is the one that was least recently used.



The system call “open” will wait for the complementary operation (write) to be opened in another process.

-> {x,y } x- minimum number of times we want to repeat; y- maximum number of times we want to repeat

-> we repeat (write) the regular expression “n” times

-> \* zero or more times

-> + one or more times

-> ? 0 or 1 time

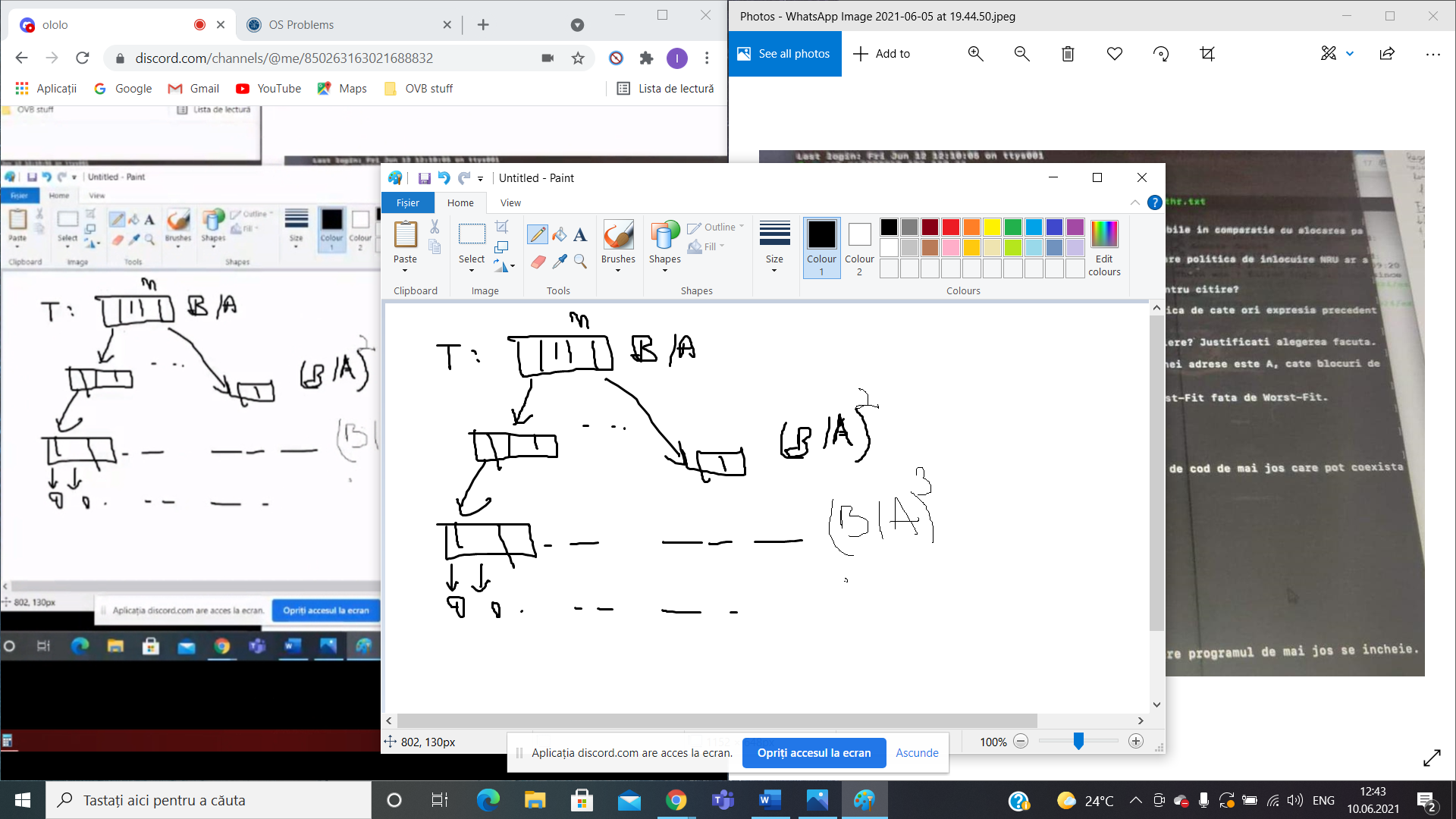
5. Cate threaduri ati folosi

-> We will use one million threads because in case of using I/O operations we can benefit from using a number of threads grater than the number of cores of the CPU because threads will enter in wait state when it comes to I/O (files) operations so while they are working on the hard disk

other threads can work on other stuff.

1. **Considerand ca dimensiunea unui bloc este B si dimensiunea unui adrese este A , cate blocuri de date sunt adresate de indirectarea tripla a unui i-nod?**

(B/A)^3



1. FIRST FIT vs WORST FIT

* AVANTAJ: First fit is faster.
* DEZAVANTAJ: There might be some small pieces of memory where it’s difficult to store other data.

1. **Cand trece un process din starea de RUN in starea de READY**

* A process will change its state from RUN to READY after he did everything he had to do on his allocated time on the processor (after its time on processor expires).