SUB 2018- 2019

1. Write a unix shell command that displays the lines in a file that

contains words starting with capital letters.

grep –E “\<[A-Z].\*\>”

grep -e '\b[A-Z][a-zA-Z]\*\b'

2. Write a unix shell command that inverts in a file b.txt all pairs of

neighboring digits

sed –r ‘s/([0-9])([0-9])/\2\1/gi’ b.txt

sed -r 's/([0-9])([0-9])/\2\1/g' input.txt

g inocuieste global, fara doar prima aparitie ar fi luata in considerare

i face sa ignore diferentele de capital (irelevant aici ca avem numere)

3. File c.txt contains on each line two numbers separated by space.

Write a unix shell command that displays for each line the sum of its

Numbers

awk ‘{$SUM=$1+$2; print $SUM}’

awk 'BEGIN {print "start"} {SUM=$1+$2; print SUM} END {print "final"}' numbers.txt

4. Display only the lines of a file d.txt that appear only once.

sort d.txt | uniq

sort d.txt |uniq

5. Write a unix shell script that displays the name of each .txt file in the

current directory, that contains the word "cat"

#!/bin/bash

for FILE in `find . –type f | grep ‘.txt$’`

do

if `grep –q ‘cat’ $FILE`

then

echo $FILE

fi

done

6. In the program fragment bellow, mark which process executes each

line: the Parent, the Child, or both.

P k = fork();

P C if (k == 0){

C printf("A\n");

}

P else {

P printf("B\n");

}

P C printf(C\n");

7. How many processes will be created by the code fragment below,

excluding the initial parent process?

fork(); wait(0); fork(); wait(0); fork();

7 p C7

C1 C5

C2 C4 C6

C3

8. What are the possible console outputs of the following code fragment

(ignoring any output that execl might generate), and when will they

happen?

printf("A\n"); execl(....); printf("B\n");  
1 A

B --- In cazul in care execl esueaza si atunci programul va continua rularea

2 A

Output execl --- in cazul in care execl nu esueaza si “paraziteaza” restul programului

9. What does the system call "read" do when the pipe is empty?

When the pipe is empty the system call read waits either until there appears some data in the pipe or until all the writers have closed their end of the pipe and there remain no writers.

10. What does the system call "open" do before returning from opening

fifo?

The system call open waits until the opposite open is called on the same FIFO. So if one opens the file with “r” the sys call open will wait until another program opens the same FIFO with “w” and vice versa.

11. Give a reason for choosing threads over processes

Threads are faster to create and take up less memory than processes do, also it is easier to communicate between them and syncronize their execution.

12. Consider that functions "fa" and "fb" are run in concurrent threads,

what will the value of "n" be after the threads are finished? Why?

pthread\_mutex\_t a, b;

int n = 0;

void\* fa(void\* p){

pthread\_mutex\_lock(&a);

n++;

pthread\_mutex\_unlock(&a);

}

void\* fb(void\* p){

pthread\_mutex\_lock(&b);

n++;

pthread\_mutex\_unlock(&b);

}

Unknown as the ++ is not an atomic operation and the functions don’t mutually exclude eachother (because they block different mutexes) while accessing the same critical resource, so the result can vary between runs.

13. Schedule the following jobs (given as Name/Duration/Deadline) so

that they all meet their deadlines: A/5/9, B/7/13, C/1/10

ACB / CAB

14. Give one advantage and one disadvantage of the segmented

allocation method over paged allocation method

AVANTAJ:

Alocarea segmentata protejeaza accesul la date din diferite segmente, in timp ce alocarea paginata nu ofera acest beneficiu. De asemenea, alocarea segmentata nu trebuie sa faca calcule de adresa complicate similare cu cele ale alocarii paginate.

DEZAVANTAJ:

Alocarea segmentata nu rezolva fragmentara, in timp ce alocarea paginata rezolva aceasta problema.

15. When would you load into memory the pages of a program that is

being stared?

As incarca paginile conform principiului vecinatatii, adica atunci cand incarc o pagina, la nevoie, le incarc si pe cele din vecinatatea ei pentru ca este foarte posibil sa avem nevoie de ele ulterior (prefetching).

Celelalte optiuni erau sa incarcam toate paginile la inceput, sau pe fiecare la nevoie, dar asta ar fi dus fie la rulaj mai incet, fie la pornire mai inceata, fapt ce nu se justifica.

16. When does a process change state from RUN to READY?

A process changes its state from RUN to READY when the scheduler decides that the program should be done with running code, so it takes over, puts the program on READY, and gives processor to another program which goes from READY to RUN.

17. Given a unix file system configured with a block size of B bytes that

can contain A addresses, and i-nodes having S direct link, one simple

indirection link, and one triple indirection link, give the formula for the

maximum file size possible.

MAX=S\*A\*B + A\*A\*B + A^3\*A\*B

18. What happens with the data when you delete a file that has a hard

link pointing to it?

If there is another hard link pointing at the data, the data remains there, not being marked as free memory, and it can be accessed through the other hard link (also the hard link counter goes down)

If there isn’t any other hard link pointing to the data, the space where the data was stored is marked as free.

19. Give a method from preventing deadlocks.

We can always lock the access through the resources in the same orther (for example always acending).

This will prevent the circular wait condition which causes deadlocks.

This also goes for deadlocks on opening FIFO’s, or other sort of deadlocks. Always blocking and/or opening the critical resources in the same order will guarantee that circular wait is eliminated and there is no deadlock possibility.

20. What is a binary semaphore and what is the effect of its P method,

when called by multiple concurrent processes/threads?

A binary semaphore functions as mutex. So if there are multiple processes/threads trying to access its P method(wait) only one will obtain access and the others will wait at the semaphore until this one “posts” to the semaphore, signaling that it has exited the critical section. Thus, only one process/thread may access the critical section at once.