***Questions EXAM***

* Why are zombie processes problematic?

They occupied in time the processer so no other new processes can be created.

* How many processes will be created (excluding the parent processes)?

1. if (fork () || fork ()) then

fork () Answer: 4 (short-circuit!!!)

1. if (fork () != fork ()) then

fork () Answer: 6

* How many processes will be created when calling f (3)?

void f (int n) {

if (n > 0 || fork () == 0) {

f (n-1);

exit (0);

}

wait (0)

}

Answer: infinite loop

* Draw the hierarchy of processes for the code:

int p = 0;

for (i = 0; i < 3; i++ ) {

if (p == 0)

p = fork ();

else

wait(0);

}

Answer: P - C1 - C2 - C3 - C4

* What will be written in the out after running this code:

Char\* s[3] = {“A”, “B”, “C”};

for (i = 0; i < 3; i++ ) {

if (fork () != 0)

execl(“/bin/echo”, “/bin/echo”, s[i], NULL);

}

Answer: A (enter) B (enter) C (enter)

* What does the system call “open” do before returning from opening a FIFO?

It waits until the FIFO is open complementary (for read and for write) to prevent deadlocks.

* Consider the producer-consumer problem with a buffer of capacity N. How many semaphores would you use to insure op correctness and what would be the semaphores initial values?

2 semaphores: one initialized with 0 and one with n.

* What does the system call “read” do when the pipe contains less data than it is required to read, but it is not empty?

It reads existing data.

* Give an advantage and a disadvantage of the set-associative caches versus the direct caches.

D: Set-associative does not use all available cache lines.

A: Set-associative avoids collision, is flexible.

* What page has the highest priority in the LRU replacement policy, when choosing a victim page?

The page with the minimum sum (least number of 1’s on its line).

* How many data blocks can be referenced to by the triple-indirection of an i-node if a block contains N addresses to other blocks?

N^3

* Give a method for preventing the apparition of deadlock, in a situation when you cannot avoid modifying resources concurrently.

Pick an order for the resources and always lock in that order.

* What is a binary semaphore, and what is the effect of its p method, when called by multiple concurrent processes/threads?

A binary semaphore has only two values 1 and 0. If value is 0, process is put in queue, else is run.

* What happens with a process between the moment it finishes and the moment its parent calls wait?

The system keeps the processes in a zombie state (it is shown in the process list but it does not execute anything).

* What does the system call “read” do when the FIFO contains less data than it is required to read, but it is not empty?

It reads existing data.

* How many data blocks can be referenced to by the double-indirection of an i-node if a block contains N addresses to other blocks?

N^2/ N - 1

* What will the code print:

int r, w, n = 0;

r = open (“abc”, O\_RDONLY);

n ++;

r = open (“abc”, O\_WRONLY);

n ++;

print (n)

Answer: 3

* What page category has the highest priority in the NRU replacement policy, when choosing a victim page?

The page with both bits 0 (not read not write)

* What does the system call “read” do when the pipe is empty?

Waits for the writer to write something or for the writer to be disconnected.

* Give a reason for choosing threads over processes.

Threads do not copy the whole program data so they save memory space because it creates only a stack (they have access to global variables etc.)

* Give one advantage and one disadvantage of the segmented allocation method over the paged allocation method.

A: Reusable segment and protected memory

D: Can cause fragmentation

* When does a process change state from RUN to READY?

When the process has been used, but the time assigned to the process is done and another one takes place.

* When does a process change state from RUN to WAIT?

When the process must wait for I/O or other resource.

* When does a process change state from RUN to SWAP?

When the processer is full and the process has been chosen as a victim.

* Explain why the parent process can't use a pipe, which has been created in the child process.

The parent does not have access to the code written in the child process.

* When would you load into memory the pages of a program, that is being started?

I would load the next pages when accessing one (is very likely that I would need those also) (Locality principle).

* What happens with the data when you delete a file that has a hard link pointing to it?

The hard link still works and point to the data even though it has been deleted.

* How many processes will be created? (do not count the root process)

Fork (); fork (); fork (); wait (); wait (); wait ();

Answer:

* What does the system call “write” do when the pipe contains less space than it is required to write?

It writes until the pipe is full even though the data does not all fit.

* What happens to a zombie process if the parent has died?

It remains in the zombie process state until it is killed.

* Cate blockuri de date pot fi referite prin tripla-indirectare a unui i-node, daca un bloc are dimensiunea B si o adresa are dimensiunea A?

(B/A) ^ 3

* Why are zombie processes a problem?

Because having multiple zombie processes will eventually fill up the processer and no other processes will have space.