

System Programming 2023

Programming HW1 Report

Name : 張閔堯

ID : B11902084

1. Busy Waiting

What it Busy Waiting

Busy waiting describes a situation that a process keep checking something and occupies the CPU resources.

How to avoid busy waiting in this assignment

Use `poll/select` to prevent busy waiting. We set the timeout of `poll/select` to be -1, which means blocking,

Therefore, we can prevent the loop that keep checking if there exists any files that can be read/write from, which wouldn't keep occupying CPU resource.

Busy waiting with `poll/select`

Yes.

If the timeout is very small or even zero, it means that the process would check the accessibility of files in a very short period of time, which means it would occupy the CPU resource frequently, which is the same as busy waiting.

2. Starvation

What is starvation

Starvation describes a situation that some requests from other processes may be waiting for a long time because the process is busy handling some other requests (Maybe the requests are from some processes with higher priority).

Possibility for starvation in this assignment

Yes.

For one server, if we keep dealing with some certain clients' request, and they just keep sending requests. For example, client1 send request at 0s, and client2 send request at 0.2s, and client1 send request in 0.4s... and dealing each request requires 0.2 seconds.

If we just keep checking from client1 to clients with larger indices, we would keep dealing with client1 and client2.

Other clients' requests wouldn't be dealt, this is starvation.

3. Maintain consistency with one process

When there's only one process, we need to record what part of the file is being written, waiting for input, or cannot be overwritten.

If one request want to write a certain part of the file, we have to make sure this part can be overwritten. (For example, check the priority)

With this method, we can make sure that no requests would cause inconsistency, because we have made sure that no requests would overwrite others' requests without checking.

4. Maintain consistency with multiple process

Because different processes may not be able to communicate to each other, we can use file lock to maintain its consistency.

For example, if a process want to deal with a part of a file, we give it `write lock`, so if other process want to access this part, it wouldn't be able to access it because `write lock` couldn't be given to two processes.

' Combining with the method mentioned in question 3, we can make sure that no processes would cause inconsistency, because we have made sure that no processes would overwrite the files without checking.