LABORATORY 5

THREADS – FORK

fork(); //CODE EXAMPLE

int pid = fork();

if(pid==0)

* You are inside of the child process;

Else

* You are inside the parent.

When the child finishes, he has to return – function exit(0);. It will return smth to parent function that will be captured by wait(0) function.

!!! for every child there has to be an exit! Otherwise: zombie process. It’s also caught in wait(), do not forget!!!

Everything that is before fork – executed once.

Everything that is after fork – a new child process is created; it has all the previously created variables and so on. The code beneath the fork is performed by both the parent and the child, if we do not use exit wait and so on.

If(pid==0){

//child shit

Exit();

}

Else{

//parent stuff

Wait();

}

For each cpu core – one process runs on it (or none).

All process run for a few miliseconds, than they are moved onto a queue and others are run and so on. We are ‘tricked’ to believe that more than x processes are run at the same time. ( x = nr of cores )