27.02.2013

İ.T.Ü.
Computer Engineering Department

BLG 312E – Computer Operating Systems
Homework 1

Submission Deadline: 13.03.2013, **Wednesday**, up to **17.00**.

- Homeworks submitted after the deadline will NOT be accepted.
- You are required to submit all 3 homeworks. Please note that you have to achieve at least 20 out of 100 points on a homework for its submission to be accepted. Homeworks with lower grades will NOT be considered as submitted. Submitting parts of the codes provided in class will NOT be sufficient to achieve a grade of 20.
- You are expected to work individually on all exams and homeworks. All forms of collaboration are discouraged and will be treated as cheating.

Homework Submission Place: You should leave your homework papers in the "Computer Operating Systems – Homework 1" box in Secretary Office of the Computer Engineering Department.

Homework Submission Type: Homework submissions should be in written form: including answers written on a paper.

Answer below questions and explain your answers in detail.

1. Compile and run `prog1.c`. Write down your comments about the order of the messages printed on the screen. Include your comments about the process IDs you encounter.

2. Modify `prog1.c` code in order to make the parent process print the process id of its own parent. Compile and run the modified program. Write down your comments about the process id of the parent's parent process.

3. Delete the `wait(NULL)` line in `prog1`. Compile and run the modified program. Write down your comments about the process id that child process print outs about its parent process.

4. Define a global integer variable at the beginning of `prog1.c` and assign a value to this variable before invoking the fork system call. Print out the value of this variable in both child and parent processes. Then, in the child and the parent processes, separately, assign different values to this variable. Afterwards, print out the value of this variable in both child and parent processes. Then, in the child and the parent processes, assign different values to this variable separately. Print out the value of this variable in both child and parent processes. Write down your comments about all of the values you have observed printed on the screen.

5. Define a pointer to a global integer variable at the beginning of `prog1.c` and allocate space for this pointer. Assign a value to this variable before the fork system call and

print out both the value of the pointer (address information) and the value stored in the place where pointer is referencing. Then, in the child and the parent processes, separately, assign different values to the integer space where the pointer references. In both child and parent processes, print out the value of pointers and the integers they are referencing to. Write down your comments about all of the values you have observed (for pointer values and the values corresponding to the integers they are referencing) printed on the screen.

6. Compile and run `p1.c` ve `p2.c`. Write down your comments about the outputs (printed messages on stdout) of both programs.