# L10-C-pipe

Deadline week 12

|  |  |
| --- | --- |
| **#** | **Assignment** |
| 1 | Implement the following structure of processes, in which the vertices indicate the communication between processes using pipes:    Process 1: reads from stdin lines of at most 30 chars and writes to process 2 the digits and to process 3 the letters  Process 2: prints the received input at stdout  Process 3: transforms to uppercase all receives letters and prints them to stdout |
| 2 | Implement the following structure of processes, in which the vertices indicate the communication between processes using pipes:    Process 1: read from stdin 10 bytes and write to each process 1,2 and 4  Processes 2, 3: add digits and replace letters with ‘Z’  Process 4: sort the result and print it  Process 5: print the information received by pipe |
|  |
| 3 | Implement the following structure of processes, in which the vertices indicate the communication between processes using pipes:    Process 1: write to 3 what is read from stdin  Process 2: read from 3 and replace all vowels with digits (aeiou->12345)  Process 3: read from 1 and write to 2 only alphanumeric lines |
|  |
| 4 | Implement the following structure of processes, in which the vertices indicate the communication between processes using pipes:    Process 1: read from stdin lines of at most 30 chars and write alternatively one line to each processes 2 and 3  Process 2: replace digits with 9 and write result to process 4  Process 3: replace letters with G and write result to process 4  Process 4: print sorted what it receives |
|  |
| 5 | Implement the following structure of processes, in which the vertices indicate the communication between processes using pipes:    Process 1: read from stdin and write towards 2 all lines that start with a letter, writes to 3 all that start with a digit and to 4 the rest of the lines  Processes 2, 3, 4: print what they read from pipe |
|  |
| 6 | Implement the following structure of processes, in which the vertices indicate the communication between processes using pipes: |
|  | Process 1: read from stdin and write to 2  Process 2: read from 1, sort and send to 4  Process 3: read from stdin, replace character X with 0 and write to 4  Process 4: print only lines with more than 10 chars |

|  |  |
| --- | --- |
| **No.** | **Assignment** |
| 7 | Implement the following structure of processes, in which the vertices indicate the communication between processes using pipes: |
|  | Process 1: read from stdin and write to 2 lines that are alphanumerical and to 3 lines that do not contain letters  Process 2: replace digits with ‘\_’  Process 3,4: print to stdout what is read |
| 8 | Parent process sends to the child a username, and the child sends back to the parent the dates when the specified user logged in. |
| 9 | Parent process sends to the child a name of a Unix server, and the child sends back to the parent the list of all usernames that work in that moment on the server. |
| 10 | Parent process sends to the child a username, and the child sends back to the parent the number of processes run by the user. |
| 11 | Parent process sends to the child a file name, and the child sends back to the parent a message that indicates the type of the file or an error message if the file does not exist. |
| 12 | Parent process sends to the child a directory name, and the child sends back to the parent the list of all text files from that directory, or an error message if the directory does not exist. |
| 13 | Parent process sends to the child a directory name, and the child sends back to the parent the total number of bytes from all files from that directory. |
| 14 | Parent process sends to the child a filename, and the child sends back to the parent the content of the file or an error message if the file does not exist. |
| 15 | Parent process sends to the child a filename, and the child sends back to the parent the number of lines from that file, or error message if the file does not exist. |
| 16 | Parent process sends to the child a filename, and the child sends back to the parent the number of words from that file. |
| 17 | Clientul ii transmite serverului un nume de fisier si un numar octal. Serverul va verifica daca fisierul respectiv are drepturi de acces diferite de numarul indicat. Daca drepturile coincid, va transmite mesajul "Totul e OK!" daca nu va seta drepturile conform numarului indicat si va transmite mesajul "Drepturile au fost modificate". |
| 18 | Clientul ii transmite serverului un nume de director iar serverul ii intoarce clientului continutul directorului indicat, respectiv un mesaj de eroare in cazul in care acest director nu exista. |
| 19 | Clientul ii transmite serverului un nume de utilizator, iar serverul ii intoarce clientului numele complet al utilizatorului si directorul personal. |
| 20 | Clientul ii transmite serverului un nume de fisier, iar serverul ii intoarce clientului numele tuturor directoarelor care contin fisierul indicat. |
| 21 | Clientul ii transmite serverului un nume de utilizator, iar serverul ii returneaza informatiile indicate de "finger" pentru utilizatorul respectiv, respectiv un mesaj de eroare daca numele respectiv nu indica un utilizator recunoscut de sistem. |