

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJASTHAN)
II SEMESTER 2021-2022

Mid Semester Test - PART A (CLOSED BOOK)

Course No.: IS F462

Course Title: Network Programming

Date: 10th March (2-3:30 PM)

Maximum Marks: 10% (10M) (35 Mins)

Note:

- Write answers in this sheet itself.
- Overwritten answers will not be accepted for rechecks
- Once you submit PART-A, you can collect PART-B.

Q1. Write answers to the following questions in the grid given below. A question may have more than one correct option. Marks will be awarded only if all correct options and only correct options are chosen.

1	2	3	4	5	6	7	8	9	10

- 1) Choose the TCP connection state(s) during connection termination on active open side.
 - (a) ESTABLISHED
 - (b) FIN_WAIT_1
 - (c) TIME_WAIT
 - (d) LAST_ACK
 - (e) SYN_SENT
- 2) The following signal(s) can be generated for a foreground process group using its terminal input
 - (a) SIGINT
 - (b) SIGSTOP
 - (c) SIGQUIT
 - (d) SIGKILL
 - (e) SIGTSTP
- 3) File descriptors
 - (a) are visible across processes
 - (b) are same file pointers returned by fopen()
 - (c) are must for any file I/O in Linux
 - (d) are always integers
- 4) dup(*fd*) system call
 - (a) duplicates *fd* from parent to child process
 - (b) duplicates entry corresponding *fd* in Open File Table
 - (c) duplicates *fd* at the highest vacant slot available file descriptor table
 - (d) duplicates *fd* at the lowest vacant slot available in file descriptor table
- 5) Following system calls can be called on any fd in Linux
 - (a) open()
 - (b) close()
 - (c) read(0)
 - (d) write()
 - (e) lseek()
- 6) Consider the following code. What will be its output?


```
main(){
    int pid=fork();
    fork();printf("hello\n");
    exit(0);
}
```

 - (a) prints hello 1 time
 - (b) prints hello 2 times
 - (c) prints hello 3 times
 - (d) prints hello 4 times
 - (e) prints hello 5 times
- 7) The following field(s) of IPv4 socket address structure require to be in Network Byte Order
 - (a) sin_family
 - (b) sin_port

- (c) `sin_addr.s_addr`
 (d) None of the above
- 8) Chose the correct option(s) about pipes.
`pipe(fds);`
 (a) `fds[0]` and `fds[1]` are available for reading
 (b) `fds[0]` is available for reading and `fds[1]` is available for writing
 (c) `fds[0]` is available for writing and `fds[1]` is available for reading
 (d) It depends on which process created pipe
 (e) it depends on `dup()` call
- 9) SYN flooding attack means
 (a) only complete queue in TCP is manipulated to become full
 (b) only incomplete queue in TCP is manipulated to become full
 (c) both complete queue and incomplete queue are manipulated to become full
 (d) Manipulating any of the two queues to become full
- 10) Writing more than `PIPE_BUF` size into a pipe during one `write()` call results in
 (a) a deadlock
 (b) interleaving of data from different `write()` calls
 (c) overwriting data of other processes
 (d) pipe `fds` getting closed by OS

Q2. Fill in the blanks. Overwritten answers will not be considered for rechecks. [10*0.5=5M]

- 1) During a `read()` on TCP socket, `read()` returns 0 when _____.
- 2) Signal which can be used for setting timers in a process is _____.
- 3) `wait()` call can be used for _____.
- 4) The purpose of set-user-id flag in File metadata is _____.
- 5) Write brief code for executing `/bin/ls` in a new process.
- 6) Write one line code for generating a signal `SIGKILL` for process 9001.
 _____.
- 7) _____ socket option can be used to abruptly end the TCP connection at application layer.
- 8) During connection establishment, TCP detects the error of service not running on remote system by _____.
- 9) A background process can't _____.
- 10) The minimum functionality of a transport layer is _____.

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJASTHAN)
II SEMESTER 2021-2022

Mid Semester Test - PART B (OPEN BOOK)

Course No.: IS F462

Course Title: Network Prog.

Date: 10th March (2-3:30 PM)

Maximum Marks: 15% (15M)

Note:

- Overwritten answers will not be accepted for rechecks
 - Write all parts of a question together.
-

Q1. Write a server program for the following requirements. Server should not use any temporary files or `system()` call or `popen()` call.

- A TCP-based client-server application has to be designed for searching a massive collection of archives stored in a single machine. A client sends a search query consisting of set of keywords up to 10 keywords in a single query. Server has to apply `grep` command separately on each keyword on all files and combine the results of all keywords together before sending to client. Server applies `grep` on all keywords not one after another but simultaneously i.e. not waiting for the previous `grep` command to finish.
- Server is to be concurrent by using `fork-per-client` model.
- A client can send multiple queries on a single TCP connection. However client waits for the reply of first query before sending the next query.
- Server has to take care of zombie processes.
- Server upon receiving signal `SIGUSR1` prints total number of clients handled so far and the number of queries handled so far.

[9M]

Q2. There is a requirement in a Linux system to receive requests from multiple users and process them according to their `userid`. A user sends a request for executing a job or program using command called *makerequest*. *makerequest* takes arguments of the path to executable file and its arguments. So any user can submit his program for execution. A program called *receiveRequests* is always running in the system. *receiveRequests* reads from a global message queue and schedules the next job to be run. The algorithm it uses is to order the jobs by the `userid` of the user who submitted them. Lower the `userid` higher the priority.

Design a solution using System V Message queues. Write programs for *makeRequest* and *receiveRequests*.

[6M]