**Important marks:**

*(incomplete)    -*  the answers marked as correct are NOT ALL correct answers.

answer            -  correct answer

answer            -  wrong answer

answer            -  you *think* it’s a correct answer

CAPS or callout  - an important note about the task.

YOU ARE ALLOWED TO ADD/UPDATE/CORRECT ANSWERS AND QUESTIONS IF YOU ARE SURE. IF YOU ARE NOT - USE YELLOW.

**In order to send an a number in range +/- 999 representing an account balance including two decimal places, between different hardware platforms and/or operating systems, with use of necessary conversions, one can use type: (edited)**

Int16\_t << wrong :// it was corrected by Borkowski

char[8]

float

Int32\_t

Int32\_t

**Which of the following data types can be safely sent and received through a network socket between different hardware platforms and/or operating systems without any conversions.**

* int32\_t
* float
* Double
* uint32\_t
* uint8\_t[4]
* uint16\_t[2]
* Int
* \*int8\_t[4]
* int8\_t[4]

**In order to send a datagram to an address given in appropriate structure, having previously opened valid socket descriptor one can (without any additional actions) call a function:**

* sendto
* fprintf
* send
* write

**In order to send a datagram to an address given in appropriate structure, having previously opened a valid socket descriptor one can (possibly with additional actions or function calls) call a function:**

* sendto
* fprintf
* send
* write

**A connect() function call will run synchronously if:*(incomplete)(edited)***

-socket's protocol is TCP, O\_NONBLOCK flag is not set, call is not interrupted by signal handling

-socket's protocol is TCP, independently of O\_NONBLOCK flag setting, disregarding previous attempts

-socket's protocol is TCP, O\_NONBLOCK flag is not set, disregarding previous attempts

-socket's protocol is UDP, independently of O\_NONBLOCK flag setting, disregarding previous attempts

**Choose UDP protocol properties:**

requires established connection of 2 network sockets

possible in-transmission data loss

no established connection, instantaneous data packets sent between sockets

Reliability

**A UDP offers easier communication than TCP in following scenarios:**

* when high reliability is required, including data ordering
* when connection context is necessary (maintaining connection state)
* when small datagrams (<500B) have to be sent atomically
* when it is necessary to send data to multiple sockets but no session maintaining is required

**A single select() function call can simultaneously:**

wait for the availability of socket to write to

wait for a signal

wait for availability of socket to read from

wait for availability of socket to read from or exclusively write to (not both at the same time)

**Optimal value passed as first argument to a pselect() call (nfds) is:**

* FD\_SETSIZE
* maximum value of all descriptors opened by process plus 1
* maximum value of all descriptors passed in descriptor sets plus 1
* 0

**The process A sends packets to the process B through a UDP socket in following order:**

**P1,P2,P3**

**Proces B can receive the following packet series:**

* P2,P1,P2,P3 (Packets can duplicate)
* P1,P1,P2,P3 (Packets can duplicate)
* P3,P2,P1
* P1,P3

**What problems may occur during transmission of datagrams using UDP?**

* Datagram duplication.
* Datagram returned to sender.
* Reordering.
* Datagram loss.

**When establishing a connection using the connect() function will run asynchronously (according to POSIX standard):**

* always, network connections are by definition asynchronous
* when the network socket was created with O\_NONBLOCK flag
* in case of interruption by signal handling (EINTR)
* if default O\_NONBLOCK flag setting for socket was not manually changed

**Correct (but not necessarily optimal) value passed as first argument to a select() call (nfds) is:**

* maximum value of all descriptors passed in descriptor sets plus 1
* 0
* maximum value of all descriptors opened by process
* FD\_SETSIZE

***A single pselect() function call can simultaneously : (incomplete)***

* wait for availability of socket to write to
* wait for availability of socket to read from or exclusively write to (not both at the same time)
* wait for availability of socket to read from
* wait for signal

**Assuming that descriptor sets passed to a select() call contain following descriptors:**

**Readfds:[ 3, 4], writefds:[4 ,5 ], errorfds: [],**

**Optimal value of nfds argument is:**

* 5
* 6 // not the only answer
* FD\_SETSIZE
* It doesn’t matter, it is forbidden to pass the same descriptor in more that one set, so the call will fail

**TCP offers easier communication than UDP in following scenarios:**

* When connection context is necessary (maintaining connection state)
* When small datagrams(<500B) have to be sent atomically
* When high reliability is required, including data ordering
* When it is necessary to send data to multiple sockets but no session maintaining is required

**Choose TCP protocol properties**

* Possible in-transmission data loss
* Requires established connection of 2 networks sockets
* No established connection, instantaneous data packets sent between sockets
* Reliability

**In order to receive a datagram having previously opened a valid socket descriptor ( possibly with additional actions or function calls) one can call function:**

* recvfrom
* fscanf
* recv
* Read

**In order to send a number in range +/- 999 999 representing an account balance including two decimal places between different hardware platforms and/or operating systems, with use of necessary conversions, one can use type:**

This one comes with different values, e.g. (+- 999 instead) This changes the char[n] size in the answer, so you gotta check that by hand. Look out for this question because answers are tricky .

* Char[11]
* Double
* Float
* Int32\_t
* Int16\_t