

Ex: NO: 7

Date: 12/9/24

Practical - 7.

Aim:

Program should achieve at least below given requirements. you can make it a bidirectional program wherein receiver is sending its data frames with acknowledgement.

Program: ~~Receiver~~ Sender.py

```
import os
```

```
def receiver (window_size, message):
```

```
    sender_buffer = "Sender-Buffer.txt"
```

```
    receiver_buffer = "Receiver-Buffer.txt"
```

```
    frame_no = 0
```

```
    frames = [[i, message[i]] for i in range(len(message))]
```

```
    while frame_no < len(frames):
```

```
        for i in range(window_size):
```

```
            if frame_no + i < len(frames):
```

```
                print(f'Sending frame: {frames[frame
```

```
                    - no + i]}
```


with open (sender - buffer , 'a') as f:

f. write (f' { frames [frame - no + i] [0] }
{ frames [frame - no + i] [1] } \n")

time. sleep (1)

while True:

if os. path. exists (receiver - buffer):

with open (receiver - buffer , 'r') as f:

ack - no = int (f. read () . strip ())

os. remove (receiver - buffer)

break.

if ack - no >= frame - no:

print (f" ACK received for frame: { ack - no }")

frame - no = ack - no + 1.

else:

print (f" NACK received for frame: { frame - no }")

resending ... ")

if __name__ == "__main__":

window_size = int (input ("Enter window size"))

message = input ("Enter message")

sender (window_size , message)

Receiver.py.

import os.

def receiver():

Sender-buffer = "Sender-Buffer.txt"

receiver-buffer = "Receiver-Buffer.txt"

expected-frame-no = 0.

while True:

if os.path.exists(Sender-buffer):

with open(Sender-buffer, 'r') as f:

lines = f.readlines()

os.remove(Sender-buffer)

for line in lines:

frame = line.strip().split()

frame-no = int(frame[0])

data = frame[1]

if frame-no == expected-frame-no:

print(f"Received frame: {frame-no},

data: {data}")

~~with open(receiver-buffer, 'w') as f:~~

~~f.write(str(frame-no))~~

~~expected-frame-no += 1.~~

else:


```
print(f'Unexpected frame: {frame-no},  
data: {data}')
```

```
with open(receiver-buffer, 'w') as f:
```

```
f.write(str(frame-no))
```

```
f.write(str(expected-frame-no))
```

```
f.write(str(expected-frame-no-1))
```

```
if __name__ == "__main__":
```

```
receiver()
```

Output: python Sender.py

Enter window size: 5

Enter message: hello.

Sending frame: [0, 'h']

Sending frame: [1, 'e']

Sending frame: [2, 'l']

Sending frame: [3, 'l']

Sending frame: [4, 'o']

NACK received for frame: 0, resending.

Sending frame: [0, 'h']

Sending frame: [4, 'e']

Sending frame: [2, 'l']

sending frame : [3, '1']

sending frame : [4, '0']

NACK received for frame : 0, resending...

sending frame : [0, 'h']

sending frame : [1, 'e']

sending frame : [2, 'l']

sending frame : [3, '1']

sending frame : [4, '0']

ACK received for frame : 4.

python receiver.py -

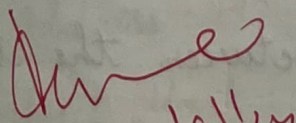
unexpected frame : 2, expected : 0

unexpected frame : 2, expected : 0.

unexpected frame : 3, expected : 0.

Result:

Thus, the program was successfully executed
& the output is verified.


18/9/24