Question Statement:
“Display network statistics information in a human readable form”

Elsa Thomas, Emmanuel Jojy, Femy Mangaly, Gary Varkey

cat /proc/net/netstat

```
TcpExt: SyncookiesSent SyncookiesRecv SyncookiesFailed EmbryonicRsts PruneCalled RcvPruned OfoPruned OutOfWindowIcmps LockDroppedIcmps ArpFilter TW
TWRecycled TWKilled PAWSActive PAWSEstab DelayedACKs DelayedACKLocked DelayedACKLost ListenOverflows ListenDrops TCPHPHits TCPPureAcks TCPHPAcks TCP
RenoRecovery TCPSackRecovery TCPSACKReneging TCPSACKReorder TCPRenoReorder TCPTSReorder TCPFullUndo TCPPartialUndo TCPDSACKUndo TCPLossUndo TCPLostR
etransmit TCPRenoFailures TCPSackFailures TCPLossFailures TCPFastRetrans TCPSlowStartRetrans TCPTimeouts TCPLossProbes TCPLossProbeRecovery TCPRenoR
ecoveryFail TCPSackRecoveryFail TCPRcvCollapsed TCPBacklogCoalesce TCPDSACKOldSent TCPDSACKOfoSent TCPDSACKRecv TCPDSACKOfoRecv TCPAbortOnData TCPAb
ortOnClose TCPAbortOnMemory TCPAbortOnTimeout TCPAbortOnLinger TCPAbortFailed TCPMemoryPressures TCPMemoryPressuresChrono TCPSACKDiscard TCPDSACKIgn
oredOld TCPDSACKIgnoredNoUndo TCPSpuriousRTOs TCPMD5NotFound TCPMD5Unexpected TCPMD5Failure TCPSackShifted TCPSackMerged TCPSackShiftFallback TCPBac
klogDrop PFMemallocDrop TCPMinTTLDrop TCPDeferAcceptDrop IPReversePathFilter TCPTimeWaitOverflow TCPReqQFullDoCookies TCPReqQFullDrop TCPRetransFail
 TCPRcvCoalesce TCPOFOQueue TCPOFODrop TCPOFOMerge TCPChallengeACK TCPSYNChallenge TCPFastOpenActive TCPFastOpenActiveFail TCPFastOpenPassive TCPFas
tOpenPassiveFail TCPFastOpenListenOverflow TCPFastOpenCookieReqd TCPFastOpenBlackhole TCPSpuriousRtxHostQueues BusyPollRxPackets TCPAutoCorking TCPF
romZeroWindowAdv TCPToZeroWindowAdv TCPWantZeroWindowAdv TCPSynRetrans TCPOrigDataSent TCPHystartTrainDetect TCPHystartTrainCwnd TCPHystartDelayDete
ct TCPHystartDelayCwnd TCPACKSkippedSynRecv TCPACKSkippedPAWS TCPACKSkippedSeq TCPACKSkippedFinWait2 TCPACKSkippedTimeWait TCPACKSkippedChallenge TC
PWinProbe TCPKeepAlive TCPMTUPFail TCPMTUPSuccess TCPDelivered TCPDeliveredCE TCPAckCompressed TCPZeroWindowDrop TCPRcvQDrop TCPWqueueTooBig TCPFast
OpenPassiveAltKey TcpTimeoutRehash TcpDuplicateDataRehash TCPDSACKRecvSegs TCPDSACKIgnoredDubious
TcpExt: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
IpExt: InNoRoutes InTruncatedPkts InMcastPkts OutMcastPkts InBcastPkts OutBcastPkts InOctets OutOctets InMcastOctets OutMcastOctets InBcastOctets Ou
tBcastOctets InCsumErrors InNoECTPkts InECT1Pkts InECT0Pkts InCEPkts ReasmOverlaps
IpExt: 0 0 0 0 1 0 229 0 0 0 229 0 0 1 0 0 0 0
```

Source Code - 1

```c
#include <stdio.h>
void display(char *a, char *b, int offset) {
    int i, j = offset;
    for (i = offset; a[i] != '\n'; i++) {
        if (a[i] == ' ') {
            printf(" -> [ ");
            while (b[j] != '\n') {
                if (b[j] == ' ') {
                    printf(" ]\n");
                    j++;
                    break;
                }
                else
                    printf("%c", b[j]);
                j++;
            }
        }
        else
            printf("%c", a[i]);
    }
    printf(" -> [ ");
    while (b[j] != '\n')
        printf("%c", b[j++]);
    printf(" ]");
}
```

Source Code - 2

```
27 void main() {
28 	char *cmd = "cat /proc/net/netstat";
29 	char tcp[2500] = " ";
30 	char tcpn[2500] = " ";
31 	char ip[2500] = " ";
32 	char ipn[2500] = " ";
33 	FILE *fp;
34 	int i, j;
35 	if ((fp = popen(cmd, "r")) == NULL) 	{
36 		printf("Error opening pipe!\n");
37 		return;
38 	}
39 	fgets(tcp, 2500, fp); fgets(tcpn, 2500, fp);
40 	printf("\n\n\n************************");
41 	printf("\nExternal TCP Statictics");
42 	printf("\n************************\n");
43 	display(tcp, tcpn, 8);
44 	fgets(ip, 2000, fp);fgets(ipn, 2000, fp);
45 	printf("\n\n\n**********************");
46 	printf("\nGeneral IP Statictics");
47 	printf("\n************************\n");
48 	display(ip, ipn, 7);
49 	printf("\n\n\n");
50 	pclose(fp);
51 	return;
52 }
```

Output

```
**********************
External TCP Statictics
**********************
SyncookiesSent  ->  [ 0 ]
SyncookiesRecv  ->  [ 0 ]
SyncookiesFailed  ->  [ 0 ]
EmbryonicRsts  ->  [ 0 ]
PruneCalled  ->  [ 0 ]
RcvPruned  ->  [ 0 ]
OfoPruned  ->  [ 0 ]
OutOfWindowIcmps  ->  [ 0 ]
LockDroppedIcmps  ->  [ 0 ]
ArpFilter  ->  [ 0 ]
TW  ->  [ 0 ]
TWRecycled  ->  [ 0 ]
TWKilled  ->  [ 0 ]
PAWSActive  ->  [ 0 ]
PAWSEstab  ->  [ 0 ]
DelayedACKs  ->  [ 0 ]
DelayedACKLocked  ->  [ 0 ]
DelayedACKLost  ->  [ 0 ]
ListenOverflows  ->  [ 0 ]
ListenDrops  ->  [ 0 ]
TCPHPHits  ->  [ 0 ]
TCPPureAcks  ->  [ 0 ]
TCPHPAcks  ->  [ 0 ]
TCPRenoRecovery  ->  [ 0 ]
TCPSackRecovery  ->  [ 0 ]
TCPSACKReneging  ->  [ 0 ]
TCPSACKReorder  ->  [ 0 ]
TCPRenoReorder  ->  [ 0 ]
TCPTSReorder  ->  [ 0 ]
```

```
TCPTSReorder  ->  [ 0 ]
TCPFullUndo  ->  [ 0 ]
TCPPartialUndo  ->  [ 0 ]
TCPDSACKUndo  ->  [ 0 ]
TCPLossUndo  ->  [ 0 ]
TCPLostRetransmit  ->  [ 0 ]
TCPRenoFailures  ->  [ 0 ]
TCPSackFailures  ->  [ 0 ]
TCPLossFailures  ->  [ 0 ]
TCPFastRetrans  ->  [ 0 ]
TCPSlowStartRetrans  ->  [ 0 ]
TCPTimeouts  ->  [ 0 ]
TCPLossProbes  ->  [ 0 ]
TCPLossProbeRecovery  ->  [ 0 ]
TCPRenoRecoveryFail  ->  [ 0 ]
TCPSackRecoveryFail  ->  [ 0 ]
TCPRcvCollapsed  ->  [ 0 ]
TCPBacklogCoalesce  ->  [ 0 ]
TCPDSACKOldSent  ->  [ 0 ]
TCPDSACKOfoSent  ->  [ 0 ]
TCPDSACKRecv  ->  [ 0 ]
TCPDSACKOfoRecv  ->  [ 0 ]
TCPAbortOnData  ->  [ 0 ]
TCPAbortOnClose  ->  [ 0 ]
TCPAbortOnMemory  ->  [ 0 ]
TCPAbortOnTimeout  ->  [ 0 ]
TCPAbortOnLinger  ->  [ 0 ]
TCPAbortFailed  ->  [ 0 ]
TCPMemoryPressures  ->  [ 0 ]
TCPMemoryPressuresChrono  ->  [ 0 ]
TCPSACKDiscard  ->  [ 0 ]
TCPDSACKIgnoredOld  ->  [ 0 ]
```

```
TCPWqueueTooBig  ->  [ 0 ]
TCPFastOpenPassiveAltKey  ->  [ 0 ]
TcpTimeoutRehash  ->  [ 0 ]
TcpDuplicateDataRehash  ->  [ 0 ]
TCPDSACKRecvSegs  ->  [ 0 ]
TCPDSACKIgnoredDubious  ->  [ 0 ]


********************
General IP Statictics
**********************
InNoRoutes  ->  [ 0 ]
InTruncatedPkts  ->  [ 0 ]
InMcastPkts  ->  [ 0 ]
OutMcastPkts  ->  [ 0 ]
InBcastPkts  ->  [ 1 ]
OutBcastPkts  ->  [ 0 ]
InOctets  ->  [ 229 ]
OutOctets  ->  [ 0 ]
InMcastOctets  ->  [ 0 ]
OutMcastOctets  ->  [ 0 ]
InBcastOctets  ->  [ 229 ]
OutBcastOctets  ->  [ 0 ]
InCsumErrors  ->  [ 0 ]
InNoECTPkts  ->  [ 1 ]
InECT1Pkts  ->  [ 0 ]
InECT0Pkts  ->  [ 0 ]
InCEPkts  ->  [ 0 ]
ReasmOverlaps  ->  [ 0 ]
```

Commands Used

- FILE *popen(char *cmd)

  System call that creates a pipe stream between the program process and the command process.

- int fgets(char *, int noOfCharacters, FILE *fp)

  To read the required number of characters from the fp and store into a character array

- int pclose(FILE *fp)

  System call that closes an open pipe.

Thank You