LAB 7

PART B

1. Leaky Bucket algorithm

#include<stdio.h>

int min(int x, int y) {

return (x < y) ? x : y;

}

int main() {

int drop = 0, mini, nsec, cap, count = 0, i, inp[25], process;

printf("Enter the bucket size: ");

scanf("%d", &cap);

printf("Enter the processing rate: ");

scanf("%d", &process);

printf("Enter the number of seconds you want to simulate: ");

scanf("%d", &nsec);

for (i = 0; i < nsec; i++) {

printf("Enter the size of the packet entering at %d sec: ", i + 1);

scanf("%d", &inp[i]);

}

printf("\n Second | Packet Received | Packet Sent | Packet Left | Dropped \n");

printf("---------------------------------------------------------------------\n");

for (i = 0; i < nsec; i++) {

count += inp[i];

if (count > cap) {

drop = count - cap;

count = cap;

}

printf("%6d | %15d |", i + 1, inp[i]);

mini = min(count, process);

printf(" %11d |", mini);

count -= mini;

printf(" %11d | %7d\n", count, drop);

drop = 0;

}

while (count != 0) {

i++;

if (count > cap) {

drop = count - cap;

count = cap;

}

printf("%6d | %15d |", i, 0);

mini = min(count, process);

printf(" %11d |", mini);

count -= mini;

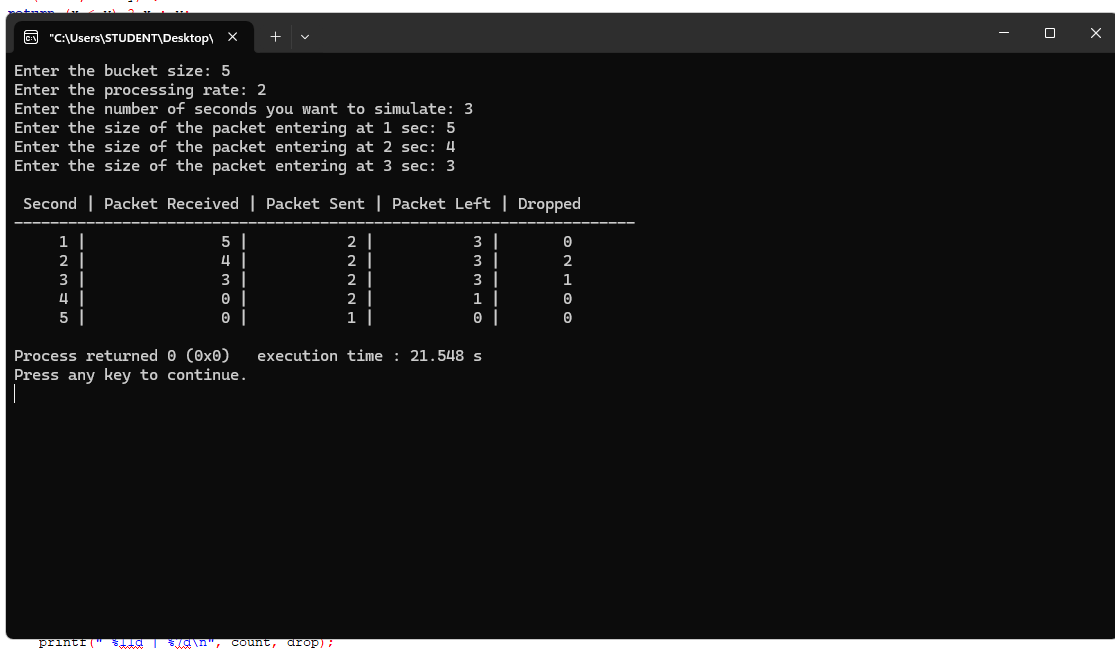
printf(" %11d | %7d\n", count, drop);

}

return 0;

}

OUTPUT:



1. CRC- CCIT

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main() {

char rem[50], a[50], s[50], c, msj[50], gen[30];

int i, genlen, t, j, flag = 0, k, n;

printf("Enter the generator polynomial: ");

fgets(gen, sizeof(gen), stdin);

gen[strcspn(gen, "\n")] = '\0';

printf("Generator polynomial is CRC: %s\n", gen);

genlen = strlen(gen);

k = genlen - 1;

printf("Enter the message: ");

n = 0;

while ((c = getchar()) != '\n') {

msj[n] = c;

n++;

}

msj[n] = '\0';

for (i = 0; i < n; i++) {

a[i] = msj[i];

}

for (i = 0; i < k; i++) {

a[n + i] = '0';

}

a[n + k] = '\0';

printf("\nMessage polynomial appended with zeros:\n");

puts(a);

for (i = 0; i < n; i++) {

if (a[i] == '1') {

t = i;

for (j = 0; j <= k; j++) {

a[t] = (a[t] == gen[j]) ? '0' : '1';

t++;

}

}

}

for (i = 0; i < k; i++) {

rem[i] = a[n + i];

}

rem[k] = '\0';

printf("The checksum appended:\n");

puts(rem);

printf("\nThe message with checksum appended:\n");

for (i = 0; i < n; i++) {

a[i] = msj[i];

}

for (i = 0; i < k; i++) {

a[n + i] = rem[i];

}

a[n + k] = '\0';

puts(a);

n = 0;

printf("Enter the received message: ");

while ((c = getchar()) != '\n') {

s[n] = c;

n++;

}

s[n] = '\0';

for (i = 0; i < n; i++) {

if (s[i] == '1') {

t = i;

for (j = 0; j <= k; j++, t++) {

s[t] = (s[t] == gen[j]) ? '0' : '1';

}

}

}

for (i = 0; i < k; i++) {

rem[i] = s[n + i];

}

rem[k] = '\0';

for (i = 0; i < k; i++) {

if (rem[i] == '1') {

flag = 1;

}

}

if (flag == 0) {

printf("Received polynomial is error-free.\n");

} else {

printf("Received polynomial has an error.\n");

}

return 0;

}

OUTPUT:

