//programmer name:sharvil Prabhudesai

//program title: process synchronisation using semaphore

#include<stdio.h>

#include<stdlib.h>

#define buffer\_size 5

int mutex=1,full=0,empty=5,in=0,out=0;

int buffer[buffer\_size];

void wait(int s);

void signal(int s );

void consumer();

void producer();

int main()

{

int choice ;

printf("process "

printf("\n producer/consumer problem \n\n");

do

{

printf("\n\n1.producer 2.consumer 3.exit");

printf("\nenter your choice :");

scanf("%d",&choice);

switch(choice)

{

case 1: producer();

break;

case 2:consumer();

break;

case 3:break;

default: printf("invalid input\n\n");

break;

}

}while(choice!=3);

return 0;

}

void producer()

{

wait(empty);

wait(mutex);

int next\_produced;

if(((in+1)%buffer\_size)==out)

{

printf("storage is full\n\n");

}

else

{

printf("enter a item to produced::");

scanf("%d",&next\_produced);

buffer[in]=next\_produced;

in=(in+1)%buffer\_size;

printf("item produced is: %d",next\_produced);

}

signal(mutex);

signal(full);

}

void consumer()

{

wait(full);

wait(mutex);

int next\_consumed;

if(in==out)

{

printf("storage is empty\n\n");

}

else

{

next\_consumed=buffer[out];

out=(out+1)%buffer\_size;

printf("item consumed is: %d",next\_consumed);

}

signal(mutex);

signal(empty);

}

void signal(int s)

{

s++;

}

void wait(int s)

{

while(s<0)

{

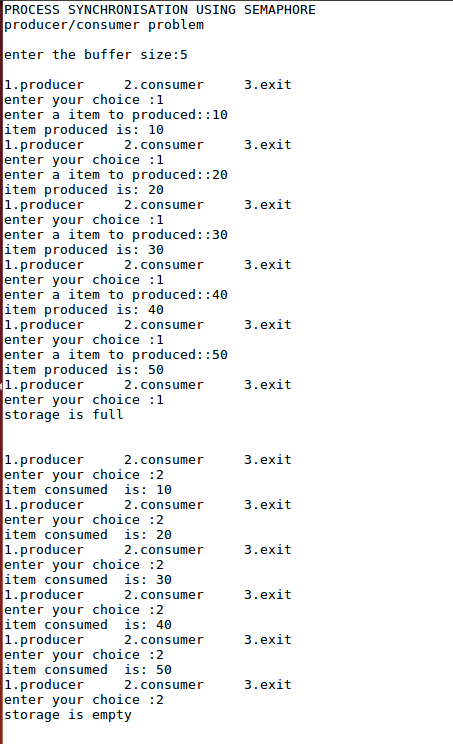
printf("cannot add an item\n\n");

exit(0);

}

s--;

}



//programmer name:sharvil Prabhudesai

//program title: process synchronisation using semaphore

#include<stdio.h>

#include<stdlib.h>

int buffer\_size;

int mutex=1,full=0,empty,in=0,out=0;

int counter=0;

int buffer[100];

void wait(int s);

void signal(int s );

void consumer();

void producer();

int main()

{

printf("PROCESS SYNCHRONISATION USING SEMAPHORE\n");

printf("producer/consumer problem \n\n");

printf("enter the buffer size:");

scanf("%d",&buffer\_size);

empty=buffer\_size;

int choice;

do

{

printf("\n1.producer 2.consumer 3.exit");

printf("\nenter your choice :");

scanf("%d",&choice);

switch(choice)

{

case 1: producer();

break;

case 2:consumer();

break;

case 3:break;

default: printf("invalid input\n\n");

break;

}

}while(choice!=3);

return 0;

}

void producer()

{

wait(empty);

wait(mutex);

int next\_produced;

if(counter==buffer\_size)

{

printf("storage is full\n\n");

}

else

{

printf("enter a item to produced::");

scanf("%d",&next\_produced);

buffer[in]=next\_produced;

in=(in+1)%buffer\_size;

printf("item produced is: %d",next\_produced);

counter++;

}

signal(mutex);

signal(full);

}

void consumer()

{

wait(full);

wait(mutex);

int next\_consumed;

if(counter==0)

{

printf("storage is empty\n\n");

}

else

{

next\_consumed=buffer[out];

out=(out+1)%buffer\_size;

printf("item consumed is: %d",next\_consumed);

counter--;

}

signal(mutex);

signal(empty);

}

void signal(int s)

{

s++;

}

void wait(int s)

{

while(s<0)

{

printf("cannot add an item\n\n");

exit(0);

}

s--;

}

