C Programs Only - OS Lab Experiments 11 to 40

Experiment 11:

```
#include<stdi o.h>

int
main()

{

int  at[ 10] ,bt[ 10] ,pr[ 10];

int

n,i,
j,t emp,t im e=0,count,over=0,sum _wait =0,sum _ turnaround=0,st a rt;

float

av gw ait ,av gturn;

print f("Ent er the numbe r of process es
\
n ");

scanf( "%d ", &n);

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

{


%d
\
n",i+1);



}

print f("Ent er

the

arrival

ti me

and

ex ecuti on

time

for
```

```
process


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

pr[ i] =i+1;
```

Experiment 12:

```
#include<stdi o.h
>

#include<conio.h>

int
main()

{

int

i,

NOP ,

sum=0,count =0,

y,

qu ant,

wt=0,

tat=0,

at[ 10] ,

bt[ 10] ,

temp[ 10] ;

float

av g_wt,

av g_tat;

print f(" Tot al number of process in the s yst em:  ");

scanf( "%d ",

&NOP );

y =

NOP ;

for(i=0;

i< NOP ;

i++)
```

```
{

print f("
\
n Ente r the
Ar rival and Bu rst ti me of the Process[ %d]
\
n", i +1);

print f("

Arrival

ti me is:

\
t ");

scanf( "%d ",

&at[ i] );
```

Experiment 13:

```
#include <stdi o.h>

#include <stdl ib.h>

#include <strin g.h>

#include <unist d.h>

#include <s
ys/i pc.h>

#include

<s ys/s hm.h>

#define S HM_S IZE 102 4

//  S iz e of the shared me mor y se gm ent

int  main() {

ke y_t

ke y

=

ftok("shmfil e",

65 );

//  Generat e

a

uni que

ke y

for

the

shar ed

memor y

se gm ent


//  C reate a ne w shar ed memor y s e gment (or get t he identifier of
an
```

```
ex ist ing
one)

int

shmi d

=

shmget(ke y,

S HM_S IZE,

IP C _CR EAT

|

0666);

if (shmid

==
-
1) {

perror ("shm get");

exit(EXIT_FA ILURE);
```

Experiment 14:

```
#include <stdi o.h>

#include <stdl ib.h>

#include <strin g.h>

#include <unist d.h>

#include <s ys/t yp es.h>

#include <s ys/i pc.h>

#include

<s ys/m s g.h>


struct m essa ge {

long

ms g_t yp e;

char

ms g_tex t[ 100] ;

};


int

main()

{

ke y_t k e y = ftok( "ms gqfi le", 65);
//  Gen erat e a uni que ke y for th e

messa ge

queue


//  C reate a ne w messa ge queue (or get t he identifi er of an ex ist ing one)

int

msgid =

msgget(k e y,

IP C _CR EAT
```

```
|

0666);

if (msgid = =
-
1) {

perror ("ms gget");
```

Experiment 15:

```
#include <stdi o.h>

#include

<pthr ead.h>


void* t hreadFun cti on(void* ar g) {

ch ar*
message = (cha r*)a r g;

pr int f("%s
\
n ",

message);
```

Experiment 16:

```
#include <stdi o.h>

#include <stdl ib.h>

#include <pthre ad.h>

#include

<unist d.h>


#define

NUM_P H ILOS OPHERS

5
```

Experiment 17:

```
#include<stdi o.h>

void bestfit(int  mp[] ,int  p[ ] ,int  m,i nt n){

int
j=0;

for(int

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

if(mp[ i] >p[ j] )
{

print f("
\
n%d

fits

in

%d ", p[ j] ,mp[ i] );

mp[ i] =mp[i]
-
p[ j++] ;

i=i
-
1;

}

}

for(int

i=j;i <m; i++)

{

print f("
\
n%d

must

wait

f or

its
```

```
process",p[ i] );
}
}


void rsort(int  a[ ] ,int  n){
for(int
i=0;i<n;i++){
for(int
j=0;j<n;j++){
if(a[ i] >a[ j] ){
int  t=a[ i] ;
a[ i] =a[ j] ;
a[ j] =t;
}
}
}
```

Experiment 18:

```
#include <stdi o.h>

#include <stdl ib.h>

#include <fcntl .h>

#include

<unist d.h>

#define BUFFER_S IZE 4
096

void
cop y( ){

const  char *sour ce file=

"C :/Users/itssk/OneDrive/Desktop/sasi.txt";

const

char

*destination_fil e="C :/Users/itssk/OneDrive/Desktop/sk.txt";

int

source_fd =

open (sourc ef il e,

O_RDON LY);

int

dest_fd

=

open(destination_file,

O_W R ONLY

|

O_CR EAT

|

O_TRUNC ,

0666);
```

```
char buf fer[ B UFFER_S IZE] ;

ssi z e_t

b ytesRe ad,

b ytesW ritten;

while

((b ytesRead

=

re ad (source_ fd,

buff er,

BU FF ER_S IZE) )

>

0)

{

b ytesW ritten

=

writ e(dest_fd,

bu ffe r,

b yt esR ead);

}

close(sourc e_fd);

close(dest_fd);

print f("Fil e

copied

suc ces sf
ull y.
\
n");

}
```

```
void

create( )

{

char

path[ 100] ;

F ILE *fp;

fp=fopen("C :/Users/itssk/OneDrive/Desktop/sasi.txt","w");

print f("file

cre ated

su cce ssfull y");

}
```

Experiment 19:

```
#include <stdi o.h>

#include

<stdl ib.h>

#include <strin g.h>

int
main() {

char mainDir ector y[ ]  = "C :/ Users/it ssk/ OneDrive/Desktop ";

ch ar

subDirector y[ ]  =

"os";

char fil e Name[ ]  = "ex am ple.tx t";

char

filePath[ 200];

char

m ainDirPath[ 200] ;

snprint
f(mainDirP ath,

siz eof(mainDirPath),

"%s/% s/",

mainDire ctor y,

subDi rector y);
```

Experiment 20:

```
#include <stdi o.h>

#include <stdl ib.h>

struct

Empl o ye e

{
```

Experiment 21:

```
#include

<stdi o.h>



#define

MAX_P R OCES S ES

5

#define MAX_RES OUR C ES 3

int
is_s afe();

int available[MAX_R ESOURC ES]  = {3, 3, 2};

//Available
inst ances of

e
a c h resourc e

int

maximum[ MAX_P ROCES S ES] [MAX_RESOUR C ES]

=

{{7,

5,

3},

{3,

2, 2},

{9,

0,

2},

{2,

2,

2},
```

```
{4,

3,

3}};

int

all ocati on[ MAX_P ROCES S ES] [MAX_RESOUR C ES]

=

{{0,

1,

0},

{2,

0, 0},

{3,

0,

2},

{2,

1,

1},

{0,

0,

2}};



int

request_resou rces (int

process_nu
m,

int

request [ ] )

{
```

```
//  C heck
if
request
can
be
granted
for
(int i
=
0;
i
<
MAX_RES OUR C ES;  i++)
{
if
(requ est[ i]
>
avail able[ i]
||
request[ i]
>
max im um[ process_num ][ i]
-
all ocati on[ process_num ] [i] )
return
0;
//  R equest cann ot be
```

```
granted

}



//

Tr y

all ocati n
g

r esourc e s

temporaril y

for (int  i = 0; i < MAX_ R ESOURC ES;  i++) {

avail able[ i]
-
= request[ i] ;
```

Experiment 22:

```
#include <stdi o.h>

#incl ude
<pthread.h >

#include
<semaphor e.h>

#include<W indows.h>


#define

BU FFER_S IZE

5

#define MAX
_ ITEMS  10

//  Max im um number of items t o be

produced/consumed



int

buffer[ BUFFER_S IZ E] ;

sem_t

empt y,

full;

int

produced_it ems

=

0,

c onsum ed_it ems

=

0;



void*
```

```
producer (void*

ar g)

{

while (produc ed_it ems <  MAX_ ITEMS ) {

sem_wait (&empt y);

//

C ritical

secti on:

add

it em

to

buffer

for (int  i = 0; i < BUFFE R _S IZE; ++i) {

if

(buffe r[ i]  == 0) {

buffer[ i]  = produced_it e ms + 1;

print f("P rodu ced:

%d
\
n",

buffer[ i] );
```

Experiment 23:

```
#include <stdi o.h>

#include

<pthr
ead.h>


//  S hared variables

int
counter

=

0;

pthread_mut ex _t

mut ex;



//  Functi on to be ex ecute d b y thre ads

void

*threadF uncti on(void

*ar g)

{
```

Experiment 24:

```
#include <stdi o.h>

#inc
l ude
<pthread.h >

#include

<semaphor e.h>


sem_t

mut ex ,

writ eBlock;

int

data

=

0,

readersCoun t =

0;



void *reade r(void *ar g) {

int
i=0;

while (i<10) {

sem_wait (&mut ex );

reade rsCount ++;

if (read ersCount  == 1 ) {

sem_wait (&write Block);

}

sem_pos t(&mut ex );



//
```

```
R eading

operati on

print f("R e ader

reads

data :

%d
\
n ",

d ata);



sem_wait (&mut ex );

reade rsCount
--
;

if (read ersCount  == 0 ) {

sem_pos t(&writ e Block );

}

sem_pos t(&mut ex );
```

Experiment 25:

```
#include

<stdi o.h>


#defineMAX_MEMORY
1000

int

memor y[ MAX_MEMORY];

//  Functi on to i nit ializ e memor y

void

ini ti aliz eMemor y( )

{

for

(int

i

=

0;

i

<

MAX_ MEMORY;

i++)

{

memor y[ i]

=

-
1;

//

-
1 indicates

that
```

```
the

memor y

is

unall ocated

}

}


//  Functi on to

displ a y me mor y status

void
displ a yMemor y( )

{

int

i, j ;

int  count

=

0;

print f( "Me mor y
```