National University

of Computer & Emerging Sciences-Islamabad
Chiniot-Faisalabad Campus

Department of Computer Science

CS205 – Operating Systems

Second Sessional Examination (Spring 2017)

Course Instructors
Rana Asif Rehman, Tayyab Javed, Rizwan Ul Haq

Monday, March 27, 2017

Total Marks: 75	**Time Allowed: 45 Minutes**

Instructions:

(1) Understanding the question paper is also part of the exam. No clarification will be offered by the instructors.
(2) Use permanent ink pens only.
(3) Exchange of stationary and possession of mobile phone is strictly disallowed.
(4) Write your name, section and complete roll number XXF-XXXX on each page.
(5) The paper consist of TWO (Objective & Subjective) parts. Objective part to be attempted in 15 minutes and subjective part to be attempted in 45 minutes. Subjective part is required to be distributed after collecting objective part.
(6) Write down the name of the author of the book to get 2 bonus marks.

Subjective Part

Roll No: _____Name: _____Section: _____

Question	1	2	3	4	5	6	7	8	Total
Points	25	5	5	5	5	5	10	15	75
Score									

Q 2. Write a code that creates 5 child processes and waits until all of the processes return. 5

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>

int main(void)
{
      int n=5, i=0, status;
      pid_t pid[5];
      bool pi=false;
      printf("Creating %d children\n", n);

      for(i=0 ; i<n ; i++)
      {
            pid[i] = fork();
            if (pid < 0) /* If an error occurred */
            {
                  printf(“Error creating child”);
                  break;
            }
            else if(pid == 0) //Stop child process to make more child processes
                  break;
            else
                  pi = true;
      }
      if(i>0 && pi) // tells that some child processes have been created.
      {
            for(int j=0 ; j < i ; j++)
                  if(d[j] > 0)
                        wait(&status);
      }
      return 0;
}
```

Q 3. What is a Zombie process? What are the benefits of having a zombie and what are the drawbacks? How can a zombie process be avoided in OS? 5

Zombie process is the one that has completed its processing but its PCB still resides in the memory because the parent process has not called **wait()** for this child process.

Benefit: Each child process will be assigned a unique process id that has not been assigned already to any other child of the parent process.

Drawback: The process might cause exhaustion of process IDs.

Q 4. Define the role of medium-term scheduler and draw the block diagram how it works? 5

It is used to lower the degree of multiprogramming. If there are too many CPU bound or I/O bound processes it swaps out partially executed process and activates the long term scheduler to bring in a new process. This way a balance of I/O and CPU bound processes ensures that all the resources are somehow being utilized equally.

Q 5. Provide a comparison between direct and indirect communication. What are the primitive operations provided by the OS to provide these inter-process communication mechanisms? 5

Direct Communication: Under direct communication, each process that wants to communicate must explicitly name the recipient or sender of the communication. In this scheme, the send() and receive() primitives are defined as:
send(P, message)—Send a message to process P.
receive(Q,message)-Receive a message from process Q

- A link is established automatically between every pair of processes that want to communicate. The processes need to know only each other's identity to communicate
- A link is associated with exactly two processes
- Between each pair of processes, there exists exactly one link

Indirect Communication: With indirect communication, the messages are sent to and received from mailboxes, or ports. A mailbox can be viewed abstractly as an object into which messages can be placed by processes and from which messages can be removed.
Each mailbox has a unique identification
The send() and receive() primitives are defined as follows:
send(A, message)—Send a message to mailbox A
receive(A, message)—Receive a message from mailbox A

- A link is established between a pair of processes only if both members of the pair have a shared mailbox
- A link may be associated with more than two processes
- Between each pair of communicating processes, a number of different links may exist, with each link corresponding to one mailbox

Q 6. How are the signals handled in multi-threaded environment? Discuss type of signals and to whom are those delivered? 5

There are two types of signals in general

1. Synchronous signal: Internal to the process (Cache miss, divide by zero etc.)

2. Asynchronous signals: External to the process (<control> <C> etc.)

A signal may be handled by one of two possible handlers:

1. A default signal handler

2. A user-defined signal handler

In general, the signal can be delivered to:

1. 1. The thread to which the signal applies.

2. 2. Every thread in the process.

3. 3. Certain threads in the process.

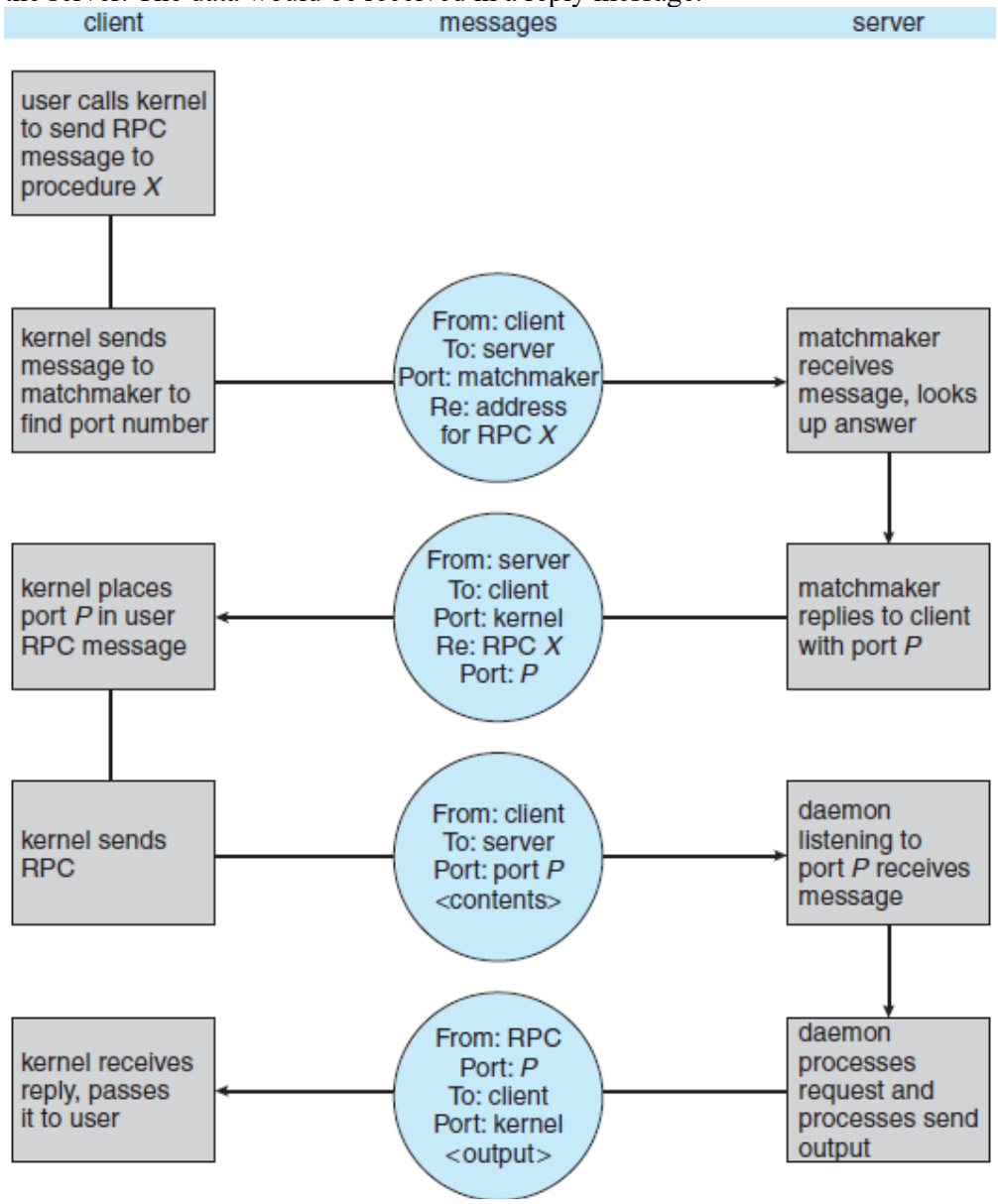
4. Assign a specific thread to receive all signals for the process

The method for delivering a signal depends on the type of signal generated

Q 7. How RPC works? Describe by the use of an example. Draw a diagram that shows the communication between client and server machine. 10

Each message is addressed to an RPC daemon listening to a port on the remote system, and each contains an identifier specifying the function to execute and the parameters to pass to that function. The function is then executed as requested, and any output is sent back to the requester in a separate message.

For instance, if a system wished to allow other systems to be able to list its current users, it would have a daemon supporting such an RPC attached to a port—say, port 3027. Any remote system could obtain the needed information (that is, the list of current users) by sending an RPC message to port 3027 on the server. The data would be received in a reply message.



Q 8. Calculate the **average waiting time** and **average turnaround time** of the processes using FCFS, SJF and SRTF for the following set of processes. Draw Gantt chart clearly. 15

Process	Burst	Arrival Time
P1	20	0
P2	25	25
P3	25	30
P4	15	40
P5	10	50
P6	10	100

FCFS:

P1		**P2**	**P3**	**P4**	**P5**	**P6**

0 20 25 50 75 90 100 110

Waiting time for
P1 = 0 – 0 = 0
P2 = 25 – 25 = 0
P3 = 50 – 30 = 20
P4 = 75 – 40 = 35
P5 = 90 – 50 = 40
P6 = 100 – 100 = 0

Average waiting time = (0+0+20+35+40+0) / 6 = 95/6 = 15.83

Turnaround time for
P1 = 20 – 0 = 20
P2 = 50 – 25 = 25
P3 = 75 – 30 = 45
P4 = 90 – 40 = 50
P5 = 100 – 50 = 50
P6 = 110 – 100 = 10

Average turnaround time = (20+25+45+50+50+10) / 6 = 200 / 6 = 33.67

Process	Burst	Arrival Time
P1	20	0
P2	25	25
P3	25	30
P4	15	40
P5	10	50
P6	10	100

SJF:

P1		**P2**	**P5**	**P4**	**P3**	**P6**

0 20 25 50 60 75 100 110

Waiting time for
P1 = 0 – 0 = 0
P2 = 25 – 25 = 0
P3 = 75 – 30 = 45
P4 = 60 – 40 = 20
P5 = 50 – 50 = 0
P6 = 100 – 100 = 0
Average waiting time = (0+0+45+20+0+0) / 6 = 65/6 = 10.83

Turnaround time for
P1 = 20 – 0 = 20
P2 = 50 – 25 = 25
P3 = 100 – 30 = 70
P4 = 75 – 40 = 35
P5 = 60 – 50 = 10
P6 = 110 – 100 = 10

Average turnaround time = (20+25+70+35+10+10) / 6 = 170 / 6 = 28.83

Process	Burst	Arrival Time
P1	20	0
P2	25	25
P3	25	30
P4	15	40
P5	10	50
P6	10	100

SRTF:

P1		**P2**	**P5**	**P4**	**P3**	**P6**

0 20 25 50 60 75 100 110

Waiting time for
P1 = 0 – 0 = 0
P2 = 25 – 25 = 0
P3 = 75 – 30 = 55
P4 = 60 – 40 = 20
P5 = 50 – 50 = 0
P6 = 100 – 100 = 0
Average waiting time = (0+0+55+20+0+0) / 6 = 75/6 = 12.5

Turnaround time for
P1 = 20 – 0 = 20
P2 = 50 – 25 = 25
P3 = 100 – 30 = 70
P4 = 75 – 40 = 35
P5 = 60 – 50 = 10
P6 = 110 – 100 = 10

Average turnaround time = (20+25+70+35+10+10) / 6 = 170 / 6 = 28.83