Total No. of Questions : 8]

SEAT No. :

P4192

[Total No. of Pages : 2

[5255]-690

M.E. (Computer Engineering)
ADVANCED UNIX PROGRAMMING
(2013 Pattern) (Credit) (Semester-III) (610102) (Elective-III)

Time : 3 Hours] *[Max. Marks : 50*

Instructions to the candidates:

1) ***Answer any five questions out of eight questions given.***
2) ***Draw neatly labeled diagram wherever necessary.***
3) ***Figures to right indicate full marks.***
4) ***Irrelevant answers will not attract any marks.***
5) ***Assume suitable data, if required.***

Q1) a) Explain how power of prediction in register stack engine help in enhancing performance IA-64 processor. **[7]**

b) With neatly labeled diagram explain IA-64 Architecture. **[3]**

Q2) a) Explain how "SIGCHILD" used in signal handling help avoiding zombie formation in UNIX system. **[7]**

b) What is signal disposition? What is the facility available to ignore unwanted signals? **[3]**

Q3) a) It possible to handle multiple inputs and output operations simultaneously in UNIX? Justify your answer in any case. **[7]**

b) What problems one may encounter if he or she used wait ( ) instead of waitpid ( ) in a program when one parent spawns multiple children which may return at different times. **[3]**

Q4) a) Write and explain program to demonstrate use of single full duplex pipe for two way communication. **[7]**

b) Explain popen ( ) and pclose ( ) calls used in PIPES. **[3]**

P.T.O.

Q5) a) Write a code and explain multithreaded server. **[7]**

b) Explain readv ( ) and writev ( ) calls. **[3]**

Q6) a) Give detail explanation of all steps involved in implementation of RPC. **[7]**

b) Why multiple threads cannot use the buffer to hold different things simultaneously. When faced with this problem, what are the various solutions available? **[3]**

Q7) a) Explain all the steps involved in implementation of concurrent server using UDP socket. **[7]**

b) State significance of SO_REUSEADDR call in socket communication. **[3]**

Q8) Write a short note on (Any Two): **[5+5]**

a) Fork ( ) with exec ( )

b) Select ( ) V/s pselect ( )

c) Close ( ) V/s shutdown ( )

d) RPC V/s RMI