**NSP-LAB2**

**Topic 2: Access to configuration files**

**Exercise 1**

Write a program in C that allows reading the services file and displaying all the services list.

|  |
| --- |
| #include <stdio.h>  int main() {  FILE \*file;  char line[256];  // Open the services file for reading  file = fopen("/etc/services", "r");  if (file == NULL) {  perror("Error opening file");  return 1;  }  // Read and display each line of the services file  printf("List of services:\n");  while (fgets(line, sizeof(line), file)) {  printf("%s", line);  }  // Close the file  fclose(file);  return 0;  } |

**Exercise:2**

Write a program in C which displays the hosts file list of a UNIX machine

|  |
| --- |
| #include <stdio.h>  int main() {  FILE \*file;  char line[256];  // Open the hosts file for reading  file = fopen("/etc/hosts", "r");  if (file == NULL) {  perror("Error opening file");  return 1;  }  // Read and display each line of the hosts file  printf("Contents of the hosts file:\n");  while (fgets(line, sizeof(line), file)) {  printf("%s", line);  }  // Close the file  fclose(file);  return 0;  } |

**Exercise:3**

1. Write a program allows us to enter the service name, and return the service description (port number and protocol) for the specific protocols (TCP and UDP), using getservbyname function.
2. Repeat the same work with the names of the services in command lines.

|  |
| --- |
| #include <stdio.h>  #include <netdb.h>  int main() {  char service\_name[50];  // Prompt the user to enter the service name  printf("Enter the service name: ");  scanf("%s", service\_name);  // Get service information for TCP protocol  struct servent \*tcp\_service = getservbyname(service\_name, "tcp");  if (tcp\_service != NULL) {  printf("TCP Service Information:\n");  printf("Service Name: %s\n", tcp\_service->s\_name);  printf("Port Number: %d\n", ntohs(tcp\_service->s\_port));  printf("Protocol: %s\n", tcp\_service->s\_proto);  } else {  printf("TCP Service not found.\n");  }  // Get service information for UDP protocol  struct servent \*udp\_service = getservbyname(service\_name, "udp");  if (udp\_service != NULL) {  printf("\nUDP Service Information:\n");  printf("Service Name: %s\n", udp\_service->s\_name);  printf("Port Number: %d\n", ntohs(udp\_service->s\_port));  printf("Protocol: %s\n", udp\_service->s\_proto);  } else {  printf("\nUDP Service not found.\n");  }  return 0;  } |

**Exercise 4**

Write a program that allows us to enter the IP address in doted-decimal notation, test if the entry address is valid, and display it in the network byte order (hexa).

|  |
| --- |
| #include <stdio.h>  #include <arpa/inet.h>  int main() {  char ip\_address[16]; // Maximum size of an IPv4 address in dotted-decimal notation is 15 characters  unsigned int ip\_network\_byte\_order;  // Prompt the user to enter an IP address  printf("Enter an IP address in dotted-decimal notation: ");  scanf("%15s", ip\_address);  // Validate the entered IP address  if (inet\_pton(AF\_INET, ip\_address, &ip\_network\_byte\_order) == 1) {  // Convert the entered IP address to network byte order (hexadecimal)  ip\_network\_byte\_order = htonl(ip\_network\_byte\_order);  // Display the IP address in network byte order (hexadecimal)  printf("IP address in network byte order (hexadecimal): 0x%X\n", ip\_network\_byte\_order);  } else {  printf("Invalid IP address.\n");  }  return 0;  } |

**Exercise 5**

Write a program in C “testport” that reads port numbers from the user keyboard and which displays if these port numbers are reserved or if they are not.

*Examples of execution of this program:*

• Test 1:

TestPort

Reply : You must give the port number

• Test 2:

TestPort 23

Reply : Port: 23 reserved for service: telnet

• Test 3:

TesterPort 26 25 80 8080 110 23

Reply :Port 26 available

Port 25 reserved for smtp service

Port 80 reserved for http service

Port 8080 is not available

Port 110 reserved for pop3 service

Port 23 reserved for telnet service

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <netdb.h>  int is\_reserved\_port(int port\_number) {  // Check if the port number falls within the reserved range  if (port\_number >= 0 && port\_number <= 1023) {  return 1; // Port number is reserved  } else {  return 0; // Port number is not reserved  }  }  void print\_service\_name(int port\_number) {  struct servent \*service\_info = getservbyport(htons(port\_number), "tcp");  if (service\_info != NULL) {  printf("reserved for service: %s\n", service\_info->s\_name);  } else {  printf("is not available\n");  }  }  int main() {  int port\_number;  // Prompt the user to enter a port number  printf("Enter a port number: ");  scanf("%d", &port\_number);  // Check if a valid port number was entered  if (port\_number <= 0) {  printf("You must give a valid port number.\n");  return 1;  }  // Check if the port number is reserved or not  if (is\_reserved\_port(port\_number)) {  printf("Port %d ", port\_number);  print\_service\_name(port\_number);  } else {  printf("Port %d available\n", port\_number);  }  return 0;  } |