LAB - 11

Page No 37
Date 9/1/23

1. Shell program on paste Cmd

```
echo "Executing paste Command"
echo "Contents of file 1"
Cat Cutlist1
echo "Contents of file 2"
Cat Cutlist2
echo "After executing the Paste Command"
paste Cutlist3 Cutlist1 Cutlist2
```

O/P

```
Contents of file 1
A
N
A
N
Y
A

Contents of file 2
S
H
E
T
T
Y
After executing the Paste Command
A    S
N    H
A    E
N    T
Y    T
A    Y
```

Note the files must be created

```
Cat > Cutlist1
A
N
A
N
Y
A
Cat > Cutlist2
S
H
E
T
T
Y
Cat > Cutlist3
```

[empty file]

2. Shell program for Sort with 3 options

```
echo "Executing of Sort Command"
sort emp.lst
echo "- - - - - - - - -"
echo "Sort of num file"
sort -n numfile1
echo "- - - - - - - - -"
echo "to Check the default file is sorted"
sort -c emp.lst
echo "- - - - - - - - -"
echo "Sorting on reverse list"
```

sort -t "|" -k emp.lst
echo

o/p Execution of sort command
0110 | v.k agrawal | g.m | marketing | 12|3|40 |
:
9876 | jai Sharma | dir | prod | 03 | 12 | 50 | 7000
Sort of num file
0
5
10
100

Sorting on reverse list
9876 | jai Sharma | dir | prd | 03 | 12 | 50 | 7000
:
0110 | v.k agrawal | g.m | marketing | 12 | 31 | 40 | 9000

3. Demo of uniq and tr cmd

uniq
o/p
cut -d "|" -f 3 emp.lst | sort | uniq -d
d.g.m
director
executive
g.m
manager

tr
o/p
echo 'translating Characters';
tr '|' '~' < emp.lst | head -n 3
2233 ~ a.k Shukla ~ g.m ~ Sales ~ 12|12|52 ~ 6000
9876 ~ jai Sharma ~ director ~ production ~ 03 b|
50| ~ 7000
5678 ~ Sumit Chakrobarty ~ d.g.m ~ marketing ~
04|19|43 ~ 6000

4. write a c pgm to that o/p's the contents of its environment list

Pgm

```
#include<stdio.h>
int main(int argc, char *argV[])
{
  int i;
  char **ptr;
  extern char **environ;
  for(ptr = environ; *ptr != 0; ptr++)
  printf("%s\n", *ptr);
  return 0;
}
```

```
gcc c1.c
./a.out
```

O/P

```
SHELL=/bin/bash
SESSION_MANAGER=local/bms6-HP-Pro-3330-MT
AT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:
XDG_MENU_PREFIX=gnome-
LANGUAGE=en_IN:en
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
```

O/P screenshot

LAB - 12

Page No. 40
Date 16/11/23

1. Write a c/c++ program to emulate the Unix command

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <string.h>
int main (int argc, char *argv[])
{
  if (argc < 3 || argc > 4 || (argc == 4 && strcmp(argv[1],
      "-s")))
  {
    printf("usage: ./a.out [-s] <org_file> <new_link>");
    return 1;
  }
  if (argc == 4)
  {
    if ((symlink (argv[2], argv[3])) == -1)
    printf(" Cannot Create Symbolic link \n")
    else
    printf (" Symbolic link Created \n"))
  }
  else
  {
    if ((link(argv[1], argv[2])) == -1)
    printf(" Cannot Create hard link \n");
    else
    printf(" Hard link Created \n");
  }
  return 0;
}
```

O/P & I/P

```
cc ln.c
./a.out 1 2 3 4
Usage: ./a.out [-s] <org_file> <new_link>
```

$ ./a.out -s 1a.c zz
Symbolic link created

2. Write a c/c++ POSIX compliant program that prints the POSIX defined configuration

```
#define _POSIX_SOURCE
_POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
_POSIX_C_SOURCE 199309L
#include <stdio.h>
#include <unistd.h>
int main()
{
#ifdef
_POSIX_JOB_CONTROL
printf("System Supports job Control \n");
#else
printf("System does not Support job Control \n");
#endif
#ifdef
_POSIX_SAVED_IDS
printf("System Supports saved set-UID and
  saved set-GID \n");
#else
printf("System does not Support saved set-
  UID and saved set-GID \n");
#endif
#ifdef _POSIX_CHOWN_RESTRICTED
printf("Chown-restricted option is %d \n",
    _POSIX_CHOWN_RESTRICTED);
#else
printf("System does not Support Chown-
  restricted option \n");
#endif
```

```
#ifdef _POSIX_NO_TRUNC
printf("Pathname trunc option is %d \n",
        _POSIX_NO_TRUNC);
#else
        printf("System does not Support System-wide pathname trunc option \n");
#endif
#ifdef _POSIX_VDISABLE
    printf("Disable Character for terminal files is
  %d \n", _POSIX_VDISABLE);
#else
printf("System does not Support _POSIX_VDISA
     -BLE \n");
#endif
return 0;
}
```

O/P

```
cc posix.c
./a.out
System supports job Control
System supports Saved Set-UID and saved
        Set-GID
Chown_restricted option is 0
Pathname trunc option is 1
Disable Character for terminal files is 0
```

3. write a c/c++ program which demonstrates interprocess communication b/w reader process and a writer process. use mkfifo, open, read, write and close API's in U'r program

```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
```

```
#include <string.h>
#include <errno.h>
#include <stdio.h>
int main (int argc, char *argv[])
{
  int fd, gt;
  char buf[256];
  if (argc != 2 && argc != 3)
  {
    printf ("USAGE %s <file> [<arg>]\n",
     argv[0]);
    return 0;
  }
  mkfifo (argv[1], S_IFIFO | S_IRWXU |
  S_IRWXG | S_IRWXO);
  if (argc == 2)
  {
    fd = open (argv[1], O_RDONLY | O_NONBLOCK);
    while (read (fd, buf, sizeof(buf)) && gt > 0)

    printf ("%s", buf);
  }
  else
  {
    fd = open (argv[1], O_WRONGLY);
    write (fd, argv[2], strlen (argv[2]));
  }
  close(fd);
}
```

O/P seq 1&1|2&3

Read writer

```
./a.out FIFO1 "This is USP &amp; CD lab"
./a.out FIFO1

This is USP
&amp; CD lab
```