

16/10/2023

~~17/10/23~~

i) Write c/c++ program to emulate the unix ln command

```
#include <stdio.h>
```

```
#include <sys/types.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] <orig-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;
}

```

→ gcc incmd.c  
 ./a.out a b  
 hard link created.  
 ./a.out -s m n  
 symbolic link created.

2) Write c/c++ POSIX compliant program that prints the POSIX defined configuration options supported on any given system using feature test macros.

```

#define _POSIX_SOURCE
#define _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 do not support")
#endif

#ifdef _POSIX_CHOWN_RESTRICTED
    printf("System supports option is %d\n", _POSIX_CHOWN_RESTRICTED);
#else
    printf("System do not support\n");
#endif

#ifdef _POSIX_NO_TRUNC
    printf("Pathname truncate option is %d\n", _POSIX_NO_TRUNC);
#endif

#ifdef _POSIX_DISABLE
    printf("Disable character for terminal files is %d\n", _POSIX_DISABLE);
#else
    return 0;
}
```

Output:

System supports job control  
System supports saved set-UID and saved set-GID

chown\_restricted option is 200119

Pathname truncate option is 200112

Disable character for terminal files is 255

```
g) #include <sys/types.h>
#include <unistd.h>
#include <sys/stat.h>
#include <string.h>
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>

int main (int argc, char* argv[])
{ int fd;
  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)))
      printf ("%s", buf);
  }
  else
  { fd = open (argv[1], O_WRONLY);
    write (fd, argv[2], strlen (argv[2]));
  }
  close (fd);
}
```

Output:

./a.out FIFO1 "This is USP &camp; CDlab"  
./a.out FIFO1  
⇒ This is USP &camp; CD lab.