UNIX SYSTEMS PROGRAMMING AND COMPILER DESIGN LABORATORY

5. a) Write a C/C++ program that outputs the contents of its Environment list

Objective : print the values of environment strings like HOME , SHELL , TERM , USER , PATH , EXINIT

Theory

- Environment Variables: have details of environment
- Form: *variable=string*
- Example: HOME=/usr1/stevens

- Environment List: Comprises list of Environment Variables.
- Array of pointers and the character strings pointed to are in data space of the process

Environment List can be accessed using:

- Environment list which are passed to *main* function as arguments
- External variable named *environ*
- Using function *getenv*

Using Environment list which are passed to *main* function

- When a program is executed , its passed a variable-length list of environment variables
- Passed as arguments – array of pointers , terminated by NULL pointer
- Are accessible to program as

```
main( argc , argv , envp )
    int argc;
    char *argv[];
    char *envp[];
        {
        }
```

Program

```
#include<iostream>
#include<stdlib.h>

using namespace std;

int main( int argc , char *argv[] , char *envp[] )
{
	for ( int i = 0 ; envp[i] != (char *) 0 ; i++ )
		{
			cout<<envp[i]<<"\n";
		}

	return 0;
}
```

Output

```
SSH_AGENT_PID=2923
HOSTNAME=ibm
DESKTOP_STARTUP_ID=
SHELL=/bin/bash
TERM=xterm
HISTSIZE=1000
GTK_RC_FILES=/etc/gtk/gtkrc:/home/rahul/.gtkrc-1.2-gnome2
WINDOWID=33554507
USER=rahul
LS_COLORS=no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;0
1:cd=40;33;01:or=01;05;37;41:mi=01;05;37;41:ex=00;32:*.cmd=00;32:*.exe=00;
32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.csh=00;32:*.tar=00;31:
*.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=
00;31:*.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;3
1:*.jpg=00;35:*.gif=00;35:*.bmp=00;35:*.xbm=00;35:*.xpm=00;35:*.png=00;35:
*.tif=00;35:
GNOME_KEYRING_SOCKET=/tmp/keyring-QxRaqe/socket
SSH_AUTH_SOCK=/tmp/ssh-hTmrVh2869/agent.2869
SESSION_MANAGER=local/ibm:/tmp/.ICE-unix/2869
USERNAME=rahul
MAIL=/var/spool/mail/rahul
DESKTOP_SESSION=default
```

PATH=/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/
home/rahul/bin
GDM_XSERVER_LOCATION=xdmcp
INPUTRC=/etc/inputrc
PWD=/home/rahul/UnixSystemProgrammingAndCompilerDesign/5a
XMODIFIERS=@im=none
LANG=en_US.UTF-8
GDMSESSION=default
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
HOME=/home/rahul
SHLVL=2
GNOME_DESKTOP_SESSION_ID=Default
LOGNAME=rahul
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-
jE6rB94YTC,guid=958918519788b239410ec2e927a65a00
LESSOPEN=|/usr/bin/lesspipe.sh %s
DISPLAY=mca28:3.0
G_BROKEN_FILENAMES=1
COLORTERM=gnome-terminal
XAUTHORITY=/tmp/.gdmQV93RW
_=./a.out
OLDPWD=/home/rahul/UnixSystemProgrammingAndCompilerDesign

Using External variable named *environ*

- External variable named *environ* can be used to access the environment list

  *extern char **environ;*

- Advantage : the environment variables need not be passed from one function to next

```
extern char **environ;

    for( int i = 0 ; environ[i] != ( char * ) 0 ; i++ )
     {
     cout<<environ[i]<<"\n";
     }
```

Using Function *getenv*

- defined in the <*stdlib.h*> header
- function allows a process to query a shell environment variable value
- Function prototype:

  *char ****getenv****(**const char*** env_name);*

```
char *env = getenv("HOME");
    cout<<"HOME=" << env << "\n";
```

Try out for :

- SHELL
- TERM
- USER
- PATH . . .

Thank you