Compiler Construction

BPDC

(Lab - 01)

1 Understand various compilation options in gcc

Consider the following code snipets.

```
/*pgm1.c*/
int a, b, c, d;
int main(void)
{
        a=(b+c)*d/b;
}
```

```
/*pgm2.c*/
int sum(int a, int b)
{//Compute sum
        return a+b;
}
int main(void)
{
        if(sum(8, 10)<20)
        {
                return -1;
        }
        return 1;
}
```

```
/*pgm3.c*/
#include <stdio.h>
#define SQR(x) (x * x)
int a = 5, b = 3, c;
int main ()
{
        c = SQR(a);
        printf("c=%\n",c);
        c = SQR(a+b);
        //function to compute square
        printf("c=%d\n", c);
}
```

Use command *man gcc* to refer to various compilation options available in gcc.

1. Run *mobaxterm* tool.
2. Access server using ssh.
3. Run *scl enable devtoolset-7 bash*
4. Run *gcc -E pgm1.c -o pgm1.i*
5. Understand the option $-E$ in the above command and analyse the same by comparing $pgm1.c$ file and $pgm1.i$.

1. Run *gcc -fdump-tree-all-graph pgm1.c -o pgm1*
2. Run *dot -Tpng pgm1.c.011t.cfg.dot -o pgm1.png*
3. Run *display pgm1.png* and analyse the output by comparing against the C file pgm1.c.

1. Run *gcc -S -m32 pgm1.c -fverbose-asm -o pgm1-p1.s*
2. Run *gcc -S -m32 -O2 pgm1.c -fverbose-asm -o pgm1-p2.s*
3. Compare the files pgm1-p1.s and pgm1-p2.s
4. Compare the RTL representation in the file pgm1.png against the code in pgm1-p1.s/pgm1-p2.s

Repeat the same using files $pgm2.c$ and $pgm3.c$.

2 Design a mini compiler

Write a program in C which takes another C program as input and checks whether the input file contains a C statement of the form $X = Y + Z$; where X, Y and Z are variables. For each statement of the above form, check whether the variables are declared. You could assume each C statement to be present in a separate line and each variable to be declared in a separate line. Further, the only supported data type is "int".