CD Lab

Lab 6
LLVM

Name: Arun Kumar Rath
SRN : PES1UG20CS076
Roll : 52
Section : B

Using LLVM Tools

main.c

```c
#include<stdio.h>

int sum(int x, int y);

int main()
{
    int r = sum(3,4);
    printf("r = %d\n", r);
    return 0;
}

```

C sum.c

```c
int sum(int x, int y)
{
    return x + y;
}
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang main.c sum.c -o sum
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ./sum
r = 7
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

Using LLVM bitcode and IR code

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang -O0 -emit-llvm -c main.c -o main.bc
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang -O0 -emit-llvm -c sum.c -o sum.bc
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ls
main.bc  main.c  sum  sum.bc  sum.c
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang -O0 -emit-llvm -S -c main.c -o main.ll
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang -O0 -emit-llvm -S -c sum.c -o sum.ll
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ls
main.bc  main.c  main.ll  sum  sum.bc  sum.c  sum.ll
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

Understanding the generated LLVM code

Exercise: Add the `-fno-discard-value-names` flag while generating human-readable code and note the differences. This flag instructs the compiler to retain variable names from the source code instead of replacing them with unnamed temporaries.

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang -O0 -emit-llvm -S -c main.c -o main.ll -fno-discard-value
-names
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang -O0 -emit-llvm -S -c sum.c -o sum.ll -fno-discard-value-n
ames
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

Converting bitcode to human readable code and vice-versa

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llvm-as main.ll -o conv_main.bc
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llvm-as sum.ll -o conv_sum.bc
```

main.ll

```
; ModuleID = 'main.c'
source_filename = "main.c"
target datalayout = "e-m:e-p270:32:32-p271:32:32-p272:64:64-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-pc-linux-gnu"

@.str = private unnamed_addr constant [8 x i8] c"r = %d\0A\00", align 1

; Function Attrs: noinline nounwind optnone uwtable
define dso_local i32 @main() #0 {
entry:
  %retval = alloca i32, align 4
  %r = alloca i32, align 4
  store i32 0, i32* %retval, align 4
  %call = call i32 @sum(i32 noundef 3, i32 noundef 4)
  store i32 %call, i32* %r, align 4
  %0 = load i32, i32* %r, align 4
  %call1 = call i32 (i8*, ...) @printf(i8* noundef getelementptr inbounds ([8 x i8], [8 x i8]* @.str, i64 0, i64 0
  ret i32 0
}

declare i32 @sum(i32 noundef, i32 noundef) #1

declare i32 @printf(i8* noundef, ...) #1

attributes #0 = { noinline nounwind optnone uwtable "frame-pointer"="all" "min-legal-vector-width"="0" "no-trappin
attributes #1 = { "frame-pointer"="all" "no-trapping-math"="true" "stack-protector-buffer-size"="8" "target-cpu"="

!llvm.module.flags = !{!0, !1, !2, !3, !4}
!llvm.ident = !{!5}

!0 = !{i32 1, !"wchar_size", i32 4}
!1 = !{i32 7, !"PIC Level", i32 2}
!2 = !{i32 7, !"PIE Level", i32 2}
!3 = !{i32 7, !"uwtable", i32 1}
!4 = !{i32 7, !"frame-pointer", i32 2}
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llvm-dis main.bc -o conv_main.ll
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llvm-dis sum.bc -o conv_sum.ll
```

Linking and converting bitcode to object files

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=asm main.bc -o main.s
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=asm sum.bc -o sum.s
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=obj main.bc -o main.o
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=obj sum.bc -o sum.o
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang main.o sum.o -o sum
/usr/bin/ld: main.o: warning: relocation in read-only section `.text'
/usr/bin/ld: warning: creating DT_TEXTREL in a PIE
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ./sum
r = 7
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llvm-link main.bc -o combined.bc
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=obj combined.bc -o combined.o
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang main.o sum.o -o sum
/usr/bin/ld: main.o: warning: relocation in read-only section `.text'
/usr/bin/ld: warning: creating DT_TEXTREL in a PIE
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ./sum
r = 7
```

Writing programs in LLVM IR

hello_world.ll

```
@.str = constant [13 x i8] c"Hello World\0A\00"
define dso_local i32 @main() {
    %printstr = getelementptr [13 x i8], [13 x i8] * @.str, i64 0, i64 0
    %1 = call i32 (i8*, ...) @printf(i8* %printstr)
    ret i32 0
}

declare i32 @printf (i8*, ...)
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llvm-as hello_world.ll -o hello_world.bc
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=obj hello_world.bc -o hello_world.o
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ./hello_world
-bash: ./hello_world: No such file or directory
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang hello_world.o -o hello_world
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ./hello_world
Hello World
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

hello_world.ll

```
@.helloworld = constant [13 x i8] c"Hello World\0A\00"
; Declare new string
@.myname = constant [20 x i8] c "My name is Chetan\0A\00"

define dso_local i32 @main() {
    %hello_world_pointer = getelementptr [13 x i8], [13 x i8] * @.helloworld, i64 0, i64
    %my_name_pointer = getelementptr [20 x i8], [20 x i8] * @.myname, i64 0, i64 0

    %1 = call i32 (i8*, ...) @printf(i8* %hello_world_pointer)
    %2 = call i32 (i8*, ...) @printf(i8* %my_name_pointer)

    ret i32 0
}

declare i32 @printf (i8*, ...)
```

namenage.ll

```
@.helloworld = constant [11 x i8] c"Age is 21\0A\00"
; Declare new string
@.myname = constant [20 x i8] c"My name is  Chetan\0A\00"

define dso_local i32 @main() {
    %hello_world_pointer = getelementptr [11 x i8], [11 x i8]* @.helloworld, i64 0, i64
    0 %my_name_pointer = getelementptr [20 x i8], [20 x i8]* @.myname, i64 0, i64 0
    %1 = call i32 (i8*, ...) @printf (i8* %my_name_pointer)
    %2 = call i32 (i8*, ...) @printf (i8* %hello_world_pointer)

    ret i32 0
}

declare i32 @printf (i8*, ...)
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llvm-as namenage.ll -o namenage.bc
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=obj namenage.bc -o namenage-o
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang namenage.o -o namenage
clang: error: no such file or directory: 'namenage.o'
clang: error: no input files
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=obj namenage.bc -o namenage.o
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang namenage.o -o namenage
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ./namenage
My name is  Chetan
Age is 21
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

fib.ll

```
@.formatstr = constant [28 x i8] c"%d Fibonacci number is: %d\0A\00"
@.elem = constant i32 5; Find 5th fibonacci number
define dso_local i32 @main() {
%elem = load i32, i32* @.elem
%fibnum = call i32 @fib (i32 %elem)
%formatptr = getelementptr [28 x i8], [28 x i8]* @.formatstr, i64 0, i64 0
%1 = call i32 (i8*, ...) @printf(i8* %formatptr, i32 %elem, i32 %fibnum)
ret i32 0
}
define dso_local i32 @fib (i32 %elem) {
%cur_val = alloca i32
%prev_val = alloca i32
%index = alloca i32
store i32 0, i32* %prev_val
store i32 1, i32* %cur_val
store i32 2, i32* %index
br label %1
1:
    %cmp_1= icmp eq i32 %elem, 1
    br i1 %cmp_1, label %2, label %3
2:
    %retval_1= load i32, i32* %prev_val
    ret i32 %retval_1
3:
    %cmp_2 = icmp eq i32 %elem, 2
    br i1 %cmp_2, label %6, label %4
4:
    %idx_1 = load i32, i32* %index
    %cmp_3 = icmp slt i32 %idx_1, %elem
    br i1 %cmp_3, label %5, label %6

```

```
5:
    %old_cur = load i32, i32* %cur_val
    %old_prev = load i32, i32* %prev_val
    %new_cur = add i32 %old_prev, %old_cur
    store i32 %new_cur, i32* %cur_val
    store i32 %old_cur, i32* %prev_val
    %idx_2 = load i32, i32* %index
    %idx_2_inc = add i32 %idx_2, 1
    store i32 %idx_2_inc, i32* %index
    br label %1

6:

    %retval_final = load i32, i32* %cur_val
    ret i32 %retval_final
}

declare i32 @printf (i8*, ...)
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llvm-as fib.ll -o fib.bc
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ llc -filetype=obj fib.bc -o fib.
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang fib.o -o fib
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ ./fib
5 Fibonacci number is: 3
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

word.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char *replaceWord (const char *s, const char *oldW, const char *newW) {
    char *result;
    int i, cnt = 0;
    int newWlen = strlen(newW);
    int oldWlen = strlen(oldW);

    for (i = 0; s [i] != '\0'; i++) {
        if (strstr (&s [i],oldW) == &s [i]) {
        cnt++;
        i += oldWlen - 1;
    }
}

result = (char *)malloc(i + cnt * (newWlen - oldWlen) + 1);
i = 0;

while (*s) {
    if (strstr (s, oldW) == s) {
        strcpy (&result[i], newW);
        i += newWlen;
        s += oldWlen;

    }
    else {
        result[i++] = *s++;
    }
}

result[i] = '\0';
return result;
```

```
}

int main() {
    char str[] = "A jellyfish stinged me on the beach.";
    char c[] = "stinged";
    char d[] = "stung";

    char *result = NULL;
    printf("Old string: %s\n", str);
    result = replaceWord (str,c,d);
    printf("New String:%s\n", result);
    free (result);
    return 0;
}
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang word.c -S -emit-llvm -Xclang -disable-O0-
optnone -fno-discard-value-names
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

word.ll

```
; ModuleID = 'word.c'
source_filename = "word.c"
target datalayout = "e-m:e-p270:32:32-p271:32:32-p272:64:64-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-pc-linux-gnu"

@__const.main.str = private unnamed_addr constant [37 x i8] c"A jellyfish stinged me on the beach.\00", align 16
@__const.main.c = private unnamed_addr constant [8 x i8] c"stinged\00", align 1
@__const.main.d = private unnamed_addr constant [6 x i8] c"stung\00", align 1
@.str = private unnamed_addr constant [16 x i8] c"Old string: %s\0A\00", align 1
@.str.1 = private unnamed_addr constant [15 x i8] c"New String:%s\0A\00", align 1

; Function Attrs: noinline nounwind uwtable
define dso_local i8* @replaceWord(i8* noundef %s, i8* noundef %oldW, i8* noundef %newW) #0 {
entry:
  %s.addr = alloca i8*, align 8
  %oldW.addr = alloca i8*, align 8
  %newW.addr = alloca i8*, align 8
  %result = alloca i8*, align 8
  %i = alloca i32, align 4
  %cnt = alloca i32, align 4
  %newWlen = alloca i32, align 4
  %oldWlen = alloca i32, align 4
  store i8* %s, i8** %s.addr, align 8
  store i8* %oldW, i8** %oldW.addr, align 8
  store i8* %newW, i8** %newW.addr, align 8
  store i32 0, i32* %cnt, align 4
  %0 = load i8*, i8** %newW.addr, align 8
  %call = call i64 @strlen(i8* noundef %0) #5
  %conv = trunc i64 %call to i32
  store i32 %conv, i32* %newWlen, align 4
  %1 = load i8*, i8** %oldW.addr, align 8
  %call1 = call i64 @strlen(i8* noundef %1) #5
  %conv2 = trunc i64 %call1 to i32
  store i32 %conv2, i32* %oldWlen, align 4
  store i32 0, i32* %i, align 4
  br label %for.cond

```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ opt -passes='module(print-callgraph)' -disable-
output word.ll
Call graph node <<null function>><<0x186c1a0>>  #uses=0
  CS<None> calls function 'replaceWord'
  CS<None> calls function 'strlen'
  CS<None> calls function 'strstr'
  CS<None> calls function 'malloc'
  CS<None> calls function 'strcpy'
  CS<None> calls function 'main'
  CS<None> calls function 'llvm.memcpy.p0i8.p0i8.i64'
  CS<None> calls function 'printf'
  CS<None> calls function 'free'

Call graph node for function: 'free'<<0x185d300>>  #uses=2
  CS<None> calls external node

Call graph node for function: 'llvm.memcpy.p0i8.p0i8.i64'<<0x1872bf0>>  #uses=1

Call graph node for function: 'main'<<0x185b400>>  #uses=1
  CS<0x186a420> calls function 'printf'
  CS<0x18619e0> calls function 'replaceWord'
  CS<0x186aa30> calls function 'printf'
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ opt -passes='module(print-callgraph)' -disable-
output namenage.ll
Call graph node <<null function>><<0x162c900>>  #uses=0
  CS<None> calls function 'main'
  CS<None> calls function 'printf'

Call graph node for function: 'main'<<0x162c9c0>>  #uses=1
  CS<0x162aa50> calls function 'printf'
  CS<0x162abc0> calls function 'printf'

Call graph node for function: 'printf'<<0x162ca40>>  #uses=3
  CS<None> calls external node

chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

Optimization using LLVM

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ opt word.ll -O3 -o word_new.ll
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ clang sum.c -S -emit-llvm -Xclang -disable-O0-o
ptnone -fno-discard-value-names
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

sum.ll

```
; ModuleID = 'sum.c'
source_filename = "sum.c"
target datalayout = "e-m:e-p270:32:32-p271:32:32-p272:64:64-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-pc-linux-gnu"

; Function Attrs: noinline nounwind uwtable
define dso_local i32 @sum(i32 noundef %x, i32 noundef %y) #0 {
entry:
  %x.addr = alloca i32, align 4
  %y.addr = alloca i32, align 4
  store i32 %x, i32* %x.addr, align 4
  store i32 %y, i32* %y.addr, align 4
  %0 = load i32, i32* %x.addr, align 4
  %1 = load i32, i32* %y.addr, align 4
  %add = add nsw i32 %0, %1
  ret i32 %add
}

attributes #0 = { noinline nounwind uwtable "frame-pointer"="all" "min-legal-vector-width"="0" "no-trapping-math"="true

!llvm.module.flags = !{!0, !1, !2, !3, !4}
!llvm.ident = !{!5}

!0 = !{i32 1, !"wchar_size", i32 4}
!1 = !{i32 7, !"PIC Level", i32 2}
!2 = !{i32 7, !"PIE Level", i32 2}
!3 = !{i32 7, !"uwtable", i32 1}
!4 = !{i32 7, !"frame-pointer", i32 2}
!5 = !{!"Ubuntu clang version 14.0.0-1ubuntu1"}
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ opt -S sum.ll -passes='mem2reg' -o sumMemReg.ll
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

sumMemReg.ll

```
; ModuleID = 'sum.ll'
source_filename = "sum.c"
target datalayout = "e-m:e-p270:32:32-p271:32:32-p272:64:64-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-pc-linux-gnu"

; Function Attrs: noinline nounwind uwtable
define dso_local i32 @sum(i32 noundef %x, i32 noundef %y) #0 {
entry:
  %add = add nsw i32 %x, %y
  ret i32 %add
}

attributes #0 = { noinline nounwind uwtable "frame-pointer"="all" "min-legal-vector-width"="0" "no-trapping-math"="true

!llvm.module.flags = !{!0, !1, !2, !3, !4}
!llvm.ident = !{!5}

!0 = !{i32 1, !"wchar_size", i32 4}
!1 = !{i32 7, !"PIC Level", i32 2}
!2 = !{i32 7, !"PIE Level", i32 2}
!3 = !{i32 7, !"uwtable", i32 1}
!4 = !{i32 7, !"frame-pointer", i32 2}
!5 = !{!"Ubuntu clang version 14.0.0-1ubuntu1"}
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ opt -S sum.ll -passes='mem2reg' -time-passes -disable-output
===-------------------------------------------------------------------------===
                      ... Pass execution timing report ...
===-------------------------------------------------------------------------===
  Total Execution Time: 0.0017 seconds (0.0017 wall clock)

   ---User Time---   --User+System--   ---Wall Time---  --- Name ---
   0.0006 ( 32.7%)   0.0006 ( 32.7%)   0.0006 ( 32.7%)  PromotePass
   0.0005 ( 31.5%)   0.0005 ( 31.5%)   0.0005 ( 31.5%)  AssumptionAnalysis
   0.0005 ( 31.3%)   0.0005 ( 31.3%)   0.0005 ( 31.3%)  TargetIRAnalysis
   0.0000 (  2.3%)   0.0000 (  2.3%)   0.0000 (  2.3%)  VerifierPass
   0.0000 (  1.9%)   0.0000 (  1.9%)   0.0000 (  1.8%)  VerifierAnalysis
   0.0000 (  0.3%)   0.0000 (  0.3%)   0.0000 (  0.3%)  DominatorTreeAnalysis
   0.0017 (100.0%)   0.0017 (100.0%)   0.0017 (100.0%)  Total

===-------------------------------------------------------------------------===
                              LLVM IR Parsing
===-------------------------------------------------------------------------===
  Total Execution Time: 0.0002 seconds (0.0002 wall clock)

   ---User Time---   --User+System--   ---Wall Time---  --- Name ---
   0.0002 (100.0%)   0.0002 (100.0%)   0.0002 (100.0%)  Parse IR
   0.0002 (100.0%)   0.0002 (100.0%)   0.0002 (100.0%)  Total
```

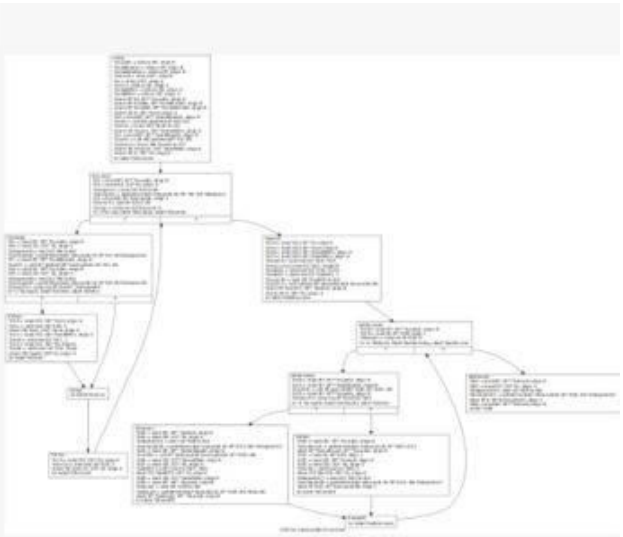
```
===-------------------------------------------------------------------------===
                              LLVM IR Parsing
===-------------------------------------------------------------------------===
  Total Execution Time: 0.0002 seconds (0.0002 wall clock)

   --System Time--   --User+System--   ---Wall Time---  --- Name ---
   0.0002 (100.0%)   0.0002 (100.0%)   0.0002 (100.0%)  Parse IR
   0.0002 (100.0%)   0.0002 (100.0%)   0.0002 (100.0%)  Total

chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ opt -S sum.ll -passes='mem2reg' -time-passes -d
isable-output -stats
===-------------------------------------------------------------------------===
                      ... Pass execution timing report ...
===-------------------------------------------------------------------------===
  Total Execution Time: 0.0018 seconds (0.0018 wall clock)

   --System Time--   --User+System--   ---Wall Time---  --- Name ---
   0.0006 ( 32.5%)   0.0006 ( 32.5%)   0.0006 ( 32.6%)  PromotePass
   0.0006 ( 31.6%)   0.0006 ( 31.6%)   0.0006 ( 31.6%)  AssumptionAnalysis
   0.0006 ( 31.5%)   0.0006 ( 31.5%)   0.0006 ( 31.5%)  TargetIRAnalysis
   0.0000 (  2.3%)   0.0000 (  2.3%)   0.0000 (  2.3%)  VerifierPass
   0.0000 (  1.8%)   0.0000 (  1.8%)   0.0000 (  1.8%)  VerifierAnalysis
   0.0000 (  0.2%)   0.0000 (  0.2%)   0.0000 (  0.2%)  DominatorTreeAnalysis
   0.0018 (100.0%)   0.0018 (100.0%)   0.0018 (100.0%)  Total

===-------------------------------------------------------------------------===
                              LLVM IR Parsing
===-------------------------------------------------------------------------===
  Total Execution Time: 0.0002 seconds (0.0002 wall clock)

   --System Time--   --User+System--   ---Wall Time---  --- Name ---
   0.0002 (100.0%)   0.0002 (100.0%)   0.0002 (100.0%)  Parse IR
   0.0002 (100.0%)   0.0002 (100.0%)   0.0002 (100.0%)  Total
```

```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ opt word.ll -passes='function(view-cfg)' -disab
le-output
Writing '/tmp/cfg.replaceWord-adfbe5.dot'...  done.
Running 'dotty' program... Remember to erase graph file: /tmp/cfg.replaceWord-adfbe5.dot
Writing '/tmp/cfg.main-7c255d.dot'...  done.
Running 'dotty' program... Remember to erase graph file: /tmp/cfg.main-7c255d.dot
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```

```
entry:
  %retval = alloca i32, align 4
  %str = alloca [37 x i8], align 1
  %c = alloca [8 x i8], align 1
  %d = alloca [6 x i8], align 1
  %result = alloca i8*, align 8
  store i32 0, i32* %retval, align 4
  %0 = bitcast [37 x i8]* %str to i8*
  call void @llvm.memcpy.p0i8.p0i8.i64(i8* align 1 %0, i8* align 1
... getelementptr inbounds ([37 x i8], [37 x i8]* @__const.main.str, i32 0, i32
... 0), i64 37, i1 false)
  %1 = bitcast [8 x i8]* %c to i8*
  call void @llvm.memcpy.p0i8.p0i8.i64(i8* align 1 %1, i8* align 1
... getelementptr inbounds ([8 x i8], [8 x i8]* @__const.main.c, i32 0, i32 0),
... i64 8, i1 false)
  %2 = bitcast [6 x i8]* %d to i8*
  call void @llvm.memcpy.p0i8.p0i8.i64(i8* align 1 %2, i8* align 1
... getelementptr inbounds ([6 x i8], [6 x i8]* @__const.main.d, i32 0, i32 0),
... i64 6, i1 false)
  store i8* null, i8** %result, align 8
  %arraydecay = getelementptr inbounds [37 x i8], [37 x i8]* %str, i64 0, i64 0
  %call = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([16 x i8],
... [16 x i8]* @.str, i64 0, i64 0), i8* %arraydecay)
  %arraydecay1 = getelementptr inbounds [37 x i8], [37 x i8]* %str, i64 0, i64
... 0
  %arraydecay2 = getelementptr inbounds [8 x i8], [8 x i8]* %c, i64 0, i64 0
  %arraydecay3 = getelementptr inbounds [6 x i8], [6 x i8]* %d, i64 0, i64 0
  %call4 = call i8* @replaceWord(i8* %arraydecay1, i8* %arraydecay2, i8*
... %arraydecay3)
  store i8* %call4, i8** %result, align 8
  %3 = load i8*, i8** %result, align 8
  %call5 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([16 x i8],
... [16 x i8]* @.str.1, i64 0, i64 0), i8* %3)
  %4 = load i8*, i8** %result, align 8
  call void @free(i8* %4) #5
  ret i32 0
```

CFG for 'main' function



```
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$ opt word.ll -passes='function(view-cfg)' -disab
le-output -cfg-func-name=replaceWord
Writing '/tmp/cfg.replaceWord-b6ed9d.dot'...  done.
Running 'dotty' program... Remember to erase graph file: /tmp/cfg.replaceWord-b6ed9d.dot
chetan@LAPTOP-ITC6BHAP:/mnt/c/College/Sem6/CD/Lab/Lab 6$
```