NEW YORK CITY COLLEGE OF TECHNOLOGY

The City University of New York

Spring, 2014

MICROCOMPUTER SYSTEM TECHNOLOGY

CET 3510L

INSTRUCTOR: Prof. Aponte

STUDENT: Yeraldina Estrella
LAB PARTNER: Odefe Enabulele

Lab 5 Procedures and Flow Control

Submission Date 4/9/2014

Laboratory Objective:

The objective of this laboratory experiment is to explore and understand the use of procedures. For the purpose of this experiment, previous lab assignments will be integrated to learn to create and call HLA procedures. A menu will be created using flow control. The users will be able to select the program they will like to run and will also have the choice to re run the program.

Analysis

In this laboratory, we have to use procedures which are another way of calling programs. The way the procedures work is we define the program in the beginning of the code. When the program will run the program will ask the user to pick which program they want to run and based on their selections the selected program will run respectably. This procedure program runs very similar to switch statements.

Program #1 procedures and Flow Control

Source code

```
program proc_flow; // creates a program name proc_flow
#include("stdlib.hhf");

procedure DecToHex(val i16:int16);  // call the program DecToHex and declare variables
begin DecToHex; // beging the program DecToHex
mov(i16,ax); // move i16 to ax
stdout.put("The decimal number is ", i16, nl); // display the decimal number
stdout.put("The Converted Hex number is $", ax, nl); // display the converted hex number
end DecToHex; // end the program

procedure DecToHexuns; // call the program DecToHexuns
static
input: uns16; // declare input to be a unsigned 16 bit integer
begin DecToHexuns; // start the program DecToHexuns
stdout.put(" ",nl);
stdout.put("Enter Decimal Number: "); // ask users to enter a decimal number
stdin.get(input); // obtain the input value and store it in the variable input
mov(input,ax); // move imput to ax
stdout.put("The decimal number is ", input, nl); // display the entered decimal number
stdout.put("The Converted Hex number is $", ax, nl); // display the converted hex number
end DecToHexuns; // end the program



procedure Adder(val i16:int16;j16:int16); // call the program Adder and declare variables
static
k16:int16; // declares k16 as integer 16

begin Adder; // start the program Adder
mov(i16,ax); // move i16 to ax
mov(j16,bx); // move j16 to bx
mov(bx,cx); // move bx to cx
add(ax,cx); // move ax to cx
mov(cx,k16);// move cs to k16
stdout.put(i16,"+",j16,"=",k16,nl); // display addition of i16 and j 16 equals to k16 (decimal number)
stdout.put(ax,"+",bx,"=",cx,nl); // display addition of ax and bx equal to cx (hex number)
end Adder; // end the program

procedure Subtractor(val i16:int16;j16:int16); // call the program Subtractor and declare variables
static
```

```
k16:int16; // declares k16 as integer 16
begin Subtractor; // start the program
mov(i16,ax); // move i16 to ax
mov(j16,bx); // move j16 to bx
mov(bx,cx); // move bx to cx
sub(ax,cx); //substract cx and ax
mov(cx,k16); // move cx into k16
stdout.put(j16,"-",i16,"=",k16,nl);
// display subtraction of j16 and i16 equals to k16 (decimal number)
stdout.put(ax,"-",bx,"=",cx,nl); // display addition of ax and bx equal to cx (hex number)
end Subtractor; // end the program substractor

procedure Multiplier(val i16:int16;j16:int16); // call the program Multiplier and declare variables
static
k16:int16; // declares k16 as integer 16
begin Multiplier; // begins the program
mov(i16,ax); // move i16 to ax
mov(j16,bx); // move j16 to bx
mov(ax,cx); // move ax to cx
mul(cx); // multiplies cx and ax
mov(ax,k16); // move ax to k16
stdout.put(j16,"*",i16,"=",k16,nl);
 // display multiplication of j16 and i16 equals to k16 (decimal number)
stdout.put(cx,"*",bx,"=",ax,nl); // display multiplication of cx and bx equal to ax (hex number)
end Multiplier; // end the program

procedure ANDOP(val i16:int16;j16:int16); // call the program ANDOP and declare variables
static
k16:int16; // declares k16 as integer 16
begin ANDOP; // begin the program
mov(i16,ax); // move i16 to ax
mov(j16,bx); // move j16 to bx
mov(ax,cx); // move ax to cx
and(ax,cx); // and operation of ax and cx
mov(ax,k16); // move ax to k16
stdout.put(j16,"&&",i16,"=",k16,nl); //display AND operation of j16 and i16 equal k16 (decimal)
stdout.put(cx,"&&",bx,"=",ax,nl); //display AND operation of cx and bx equal ax (hex)
end ANDOP; // end the program

procedure OROP(val i16:int16;j16:int16); // call the program OROP and declare variables
static
k16:int16; // declares k16 as integer 16
begin OROP; // begins the program
mov(i16,ax); // move i16 to ax
mov(j16,bx); // move j16 to bx
mov(ax,cx); // move ax to cx
or(ax,cx); // or operation of ax and cx
mov(ax,k16); // move ax to k16
stdout.put(j16,"||",i16,"=",k16,nl); //display OR operation of j16 and i16 equal k16 (decimal)
```

```
stdout.put(cx,"||",bx,"=",ax,nl); //display OR operation of cx and bx equal ax (hex)
end OROP; // end the program

procedure XOROP(val i16:int16;j16:int16); // call the program XOROP and declare variables
static
k16:int16; // declares k16 as integer 16
begin XOROP; // begins the program
mov(i16,ax); // move i16 to ax
mov(j16,bx); // move j16 to bx
mov(ax,cx); // move ax to cx
xor(ax,cx); //x or operation of ax and cx
mov(ax,k16); // move ax to k16
stdout.put(j16,"XOR",i16,"=",k16,nl); //display XOR operation of j16 and i16 equal k16 (decimal)
stdout.put(cx,"XOR",bx,"=",ax,nl); //display XOR operation of cx and bx equal ax (hex)
end XOROP; // end the program

procedure EXTENSION(val i8:int8); // call the program EXTENSION and declare variables
static
i16:int16; // declares i16 as integer 16
begin EXTENSION; // begins the program
mov(i8, al);// move i8 to al
stdout.put("You entered",i8, "($",al,")",nl); // display the entered value
cbw();//convert double word
mov(ax, i16);// move ax to i16
stdout.put("The 16-bit sign extension",i16, "($",ax,")",nl); // display the i16 sign extension
end EXTENSION; // end the program

procedure EXTENSIONUNS(val i8:int8);
// call the program EXTENSIONUNS and declare variables
static
i16:uns16; // declares i16 as unsigned integer 16
begin EXTENSIONUNS; // beging the program
stdout.put("Enter a small Postive number: "); // ask user to enter a positive number
stdin.get(i8); // obtain the input value and store it in i8
mov(i8, al);// move i8 to the al register
stdout.put("You entered",i8, "($",al,")",nl); // display the entered value
cbw();// convert a double word
mov(ax, i16);//move ax to i16
stdout.put("The 16-bit unsignedsign extension",i16, "($",ax,")",nl);
 // display the unsigned extension
end EXTENSIONUNS; // end the program




static
ch1:char:='y'; // declare ch1 as a character y
ch2:char; declare ch2 as a character
pi8:int8; // declare pi8 as a integer 8 bit
```

```
pi16:int16; // declare pi16 as a integer 16 bit
pj16:int16; // declare pj16as a integer 16 bit

begin proc_flow; // begin the procedure flow

while(ch1='Y'||ch1='y')do // while loop

Menu:
stdout.put("(a) Decimal to Hexadecimal", nl); // program a
stdout.put("(b) Decimal to Hexadecimal uns", nl); // program b
stdout.put("(c) Adder", nl); // program c
stdout.put("(d) substractor", nl); // program d
stdout.put("(e) multiplier", nl); // program e
stdout.put("(f) AND OP", nl); // program f
stdout.put("(g) OR OP", nl); // program g
stdout.put("(h) XOR OP", nl); // program h
stdout.put("(i) EXTENSIONS", nl); // program i
stdout.put("(j) EXTENSION uns", nl); // program j
stdout.put("(k) Exit", nl); // program k
stdin.get(ch2);  // obtain the ch2
stdin.flushInput(); // clear the past input

if (ch2='a') then  // if statement for input a
stdout.put("Decimal to Hexadecimal"); // display the selected program to run
call stdout.newln;
//call DecToHex();
stdout.put("input a number in decimal format",nl); // ask user to input a decimal number
stdin.get(pi16); // obtain the input and store it in pi16
DecToHex(pi16); // send pi16 to the program DecToHex


elseif (ch2='b') then // elseif statement for input b
stdout.put("Decimal to Hexadecimal");// display the selected program to run
call stdout.newln;
call DecToHexuns; // call the program

elseif (ch2='c') then // elseif statement for input c
stdout.put("Call Adder"); // display the selected program to run
call stdout.newln;
//call Adder();
stdout.put("input two number in decimal format",nl); // ask user to input two decimal numbers
stdin.get(pi16); // obtain the input and store it in pi16
stdin.get(pj16); // obtain the input and store it in pj16
Adder(pi16,pj16); // run the program adder with the input values

elseif(ch2='d') then // elseif statement for input d
stdout.put("Call Subtractor",nl);
call stdout.newln;
//stdout.put("Call Subtractor",nl);
```

```
stdout.put("input two number in decimal format",nl); // ask user to input two numbers
stdin.get(pi16); // obtain the input value and store it in pi16
stdin.get(pj16); // obtain the input value and store it in pj16
Subtractor(pj16,pi16); // run the program subtractor with the input values

elseif (ch2='e') then // elseif statement for input e
stdout.put("Call Multiplier");
call stdout.newln;
//call Mul();
stdout.put("input two number in decimal format",nl); // ask user to input two numbers
stdin.get(pi16); // obtain the input value and store it in pi16
stdin.get(pj16); // obtain the input value and store it in pj16
Multiplier(pj16,pi16); // run the program multiplier with the input values

elseif (ch2='f') then // elseif statement for input f
stdout.put("ANDOP Procedure");
call stdout.newln;
//call ANDOP();
stdout.put("input two number in decimal format",nl); // ask user to input two numbers
stdin.get(pi16); // obtain the input value and store it in pi16
stdin.get(pj16); // obtain the input value and store it in pj16
ANDOP(pi16,pj16); // run the program ANDOP with the input values

elseif (ch2='g') then // elseif statement for input g
stdout.put("OROP Procedure");
call stdout.newln;
//call OROP();
stdout.put("input two number in decimal format",nl); // ask user to input two numbers
stdin.get(pi16); // obtain the input value and store it in pi16
stdin.get(pj16); // obtain the input value and store it in pj16
OROP(pi16,pj16); // run the program OROP with the input values

elseif (ch2='h') then // elseif statement for input h
stdout.put("Call Bitwise XOR Procedure");
call stdout.newln;
//call XOROP();
stdout.put("input two number in decimal format",nl); // ask user to input two numbers
stdin.get(pi16); // obtain the input value and store it in pi16

stdin.get(pj16); // obtain the input value and store it in pj16
XOROP(pi16,pj16); // run the program XOROP with the input values

elseif (ch2='i') then // elseif statement for input i
stdout.put("Call Extension Procedure");
call stdout.newln;
//call EXTENSION();
stdout.put("input a number in decimal format",nl); // ask user to input a numbers
stdin.get(pi8); // obtain the input value and store it in pi8
EXTENSION(pi8); // run the program Extensionwith the input value
```

```
elseif (ch2='j') then // elseif statement for input j
stdout.put(" Call Extensionuns Procedure");
call stdout.newln;
call EXTENSIONUNS; // call the program


elseif(ch2='k') then // elseif statement for input j
jmp endexit; // jump
endif; // end if statement

stdout.put("",nl); // display new line
stdin.flushInput(); // clear previous input
stdout.put("Do you want to use the program again? (Y/N)", nl);
// ask user if they want to use program again
stdin.get(ch1); // obtain ch1
stdin.flushInput(); // clear the input entered
endwhile; // end the while loop

endexit: // end

stdout.put("Thank you",nl); // display  Thank You
end proc_flow; // end the program
```

Lab Results / Outputs Results

Program #1 Procedures and Flow Control

Output Dec To Hex

input a number in decimal format
10
The decimal number is 10
The Converted Hex number is $000A

Output Dec to Hex Uns

Enter Decimal Number: 11
The decimal number is 11
The Converted Hex number is $000B

Output for Adder

input two number in decimal format
10
5
10+5=15
000A+0005=000F

Output for Subtractor

input two number in decimal format
20
15
20-15=5
000F-0014=0005

Output for Multiplier

input two number in decimal format
5
4
5*4=16
0004*0005=0010

Output for AND OP

input two number in decimal format
10
5
5&&10=10
000A&&0005=000A

Output for OR OP

input two number in decimal format
10
3
3||10=10
000A||0003=000A

Output for XOR

input two number in decimal format
10
15
15XOR10=10
000AXOR000F=000A

Output for Extensions

input a number in decimal format
10
You entered10($0A)
The 16-bit sign extension10($000A)

Output for Extensions Uns

Enter a small Postive number: 10
You entered10($0A)
The 16-bit unsigned sign extension10($000A)

Conclusion

In conclusion, the program worked accordingly, we were able to call the different procedures and run them. It worked flawlessly just like using a switch case statement.