

Laboratory 05

Stack

CO 2103 Assembly Language

Objective

Investigate Stack operations

- investigate PUSH & POP using DEBUG
- investigate CALL & RET using DEBUG
- recursion example – factorial program

PUSH & POP

- **Task 1:** Complete the program in `pushpop.txt` (all numbers in hex), save it as `pushpop.asm` and create its `EXE` file using `MASM` & `LINK`.
- **Task 2:** Load the `EXE` file in `DEBUG` and examine the program as per comments in the file.
 - use `u` to display your program
 - use `t` to step through the instructions
 - use `d` to display the content of Stack (memory) – taking note `SS` and `SP`

CALL & RET

- **Task 3:** Complete the program in `callret.txt` (all numbers in hex), save it as `callret.asm` and create its `EXE` file using `MASM` & `LINK`.
- **Task 4:** Load the `EXE` file in `DEBUG` and examine the program as per comments in the file.

Recursion example: the Factorial Program

- **Task 5:** Investigate the `fact1.asm` program in **DEBUG**:
 - try with different number for `AX` and confirm that the program performs factorial (trace the program to see the result in `AX`)
 - use a small value for `AX`, say `5`, trace and record the content of `CS`, `DS`, `IP`, `SP`, `BP`, `AX` and `Stack` after every `CALL` and `RET`
 - explain every instruction in the program
 - observe how parameters/results are passed
 - what do you think is the purpose of `BP`?