Sarbani Bhattacharyya, A20383075, Chen En Lee, A20435695

Programming Assignment 3

README

FILE ORGANIZATION

- **src/**
  - **part1/ :** This folder contains the code pertaining to Part 1 of PA3.
    - **Makefile:** Enables user to type *make -B* into the command line and compile all the C files.
    - **Memleak.c:** Corresponds to our response to Question 1
    - **test1.c:** Valid test case, demonstrates correct memory allocation and freeing
    - **test2.c:** Invalid test case, demonstrates allocating memory but not freeing
    - **test3.c:** Invalid test case, demonstrates allocating memory but not freeing
    - **test4.c:** Invalid test case: shows reading a memory location after freeing it
    - **test5.c:** Invalid test case: shows writing to memory location after freeing it
    - **test6.c:** Invalid test case: shows allocating insufficient memory and then writing to it.
    - **test7.c:** Invalid test case: Indirectly lost memory example
    - **test8.c:** Invalid test case: Possibly lost memory example
    - **test9.c:** Invalid test case: Memory leak due to use of uninitialized values
    - **test10.c:** Invalid test case: Changing the allocated pointer's address to NULL
    - **test11.c:** Invalid test case: Local pointer obtained from called function used in the caller function
    - **test12.c:** Invalid test case: Changing the pointer's address to NULL, example using array

  - **part2/ xv6-public:** This folder contains the code pertaining to Part 2 of PA3.
    - **Proc.c:** Contains system calls "myV2p" and "hasPage".
    - **pageTest.c:** Test cases
    - **v2pTest.c:** Test cases.

- **doc/ :** Contains the documentation that explains what our code is doing.
  - **part1/ :** Answer for part1
    - *PA3 Part 1 Documentation.pdf*: PDF file containing the results of Part 1 of PA3.

  - **part2/ :** Answer for part2
    - *Part 2_Documentation:* The design and manPage of Part 2 of PA3.
    - TestFile: The test result of Part 2 of PA3.

- **README:** PDF File that shows how to run our code, and explains how our files are arranged.

Sarbani Bhattacharyya, A20383075, Chen En Lee, A20435695

PART 1

Steps to run code pertaining to Part 1 of PA3.
NOTE: You must have ***valgrind*** and ***gcc*** already set up in your machine.

Step 1: *cd* into *src/part1* folder from assignment root directory:

```
cd src/part1
```

Step 2: Compile memleak.c and all the 12 test C source files using the following command:

```
make -B
```

Step 3: Run the valgrind test for each .c file using the following command

```
valgrind --leak-check=yes ./<source filename>
```

Examples:

```
valgrind --leak-check=yes ./memleak //runs memleak.c
  valgrind --leak-check=yes ./test1 //runs test1.c
  valgrind --leak-check=yes ./test2 //runs test2.c
                          .
                          .
                          .
 valgrind --leak-check=yes ./test12 //runs test12.c
```

Sarbani Bhattacharyya, A20383075, Chen En Lee, A20435695

PART 2

Steps to run code pertaining to Part 2 of PA3.

Step 1: cd into the xv6 directory:

```
cd src/part2/xv6-public/
```

Step 2: Running the following command will give us a xv6 shell

```
make qemu-nox
```

Step 3: Run the v2p test,

```
v2pTest
```

Step 4: Run the hasPage Test

```
pageTest
```