

Assignment 1

Total: 100 points

Due Date: Sunday, September 24, 2017, 11:59pm

Submit electronically on iLMS

What to submit: One zip file named `<studentID>-hw1.zip` (replace `<studentID>` with your own student ID). It should contain two files:

- one PDF file for Section 1. Write your answers in English. Check your spelling and grammar. Include your name and student ID!
- one Python source file named "hw1.py" (exact upper and lower case) for the function `PrintBST()` in Section 3. Include your name and student ID in the program comments on top.

1. Problem Set

Read Chapter 1 of Sieberchatz 9th edition book. Then, do the following problems.

1. [25 points] Which of the following instructions should be privileged for an OS to function properly? Explain.
 - a. Set value of timer.
 - b. Read the clock.
 - c. Issue a trap instruction.
 - d. Turn off interrupts.
 - e. Access I/O device.
2. [25 points] (Problem 1.8) What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?
3. [25 points] (Problem 1.12) Consider an SMP system similar to the one shown in Fig. 1.6. Illustrate with an example how data residing in memory could in fact have a different value in each of the local caches. [Hint: all you need to do is to explain an example of how this can happen]

2. Setup VMware (or VirtualBox)

- Download one of the images for Ubuntu Linux. (no need to download both)
 - VMware: (preferred: VMware player is free for Windows) 1.4 GB of disk image
<https://drive.google.com/file/d/0B-muRhFOI1UTd01kT3g2Rm15MUE/view?usp=sharing>

- VirtualBox: (similar size)

<https://drive.google.com/file/d/0B-muRhFOI1UTeGhCV19sbjdWdlU/view?usp=sharing>

- Start the Linux on the VM. Login with user name: nthu-os, password: nthu
- Go to the directory {Nachos}/code/test
- `make clean`
- `make`
- `../build.linux/nachos -e halt`

You should get the following output from running Nachos:

```
halt
Machine halting!
```

```
This is halt
Ticks: total 52, idle 0, system 40, user 12
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
```

If you cannot get to this point, ask for help immediately.

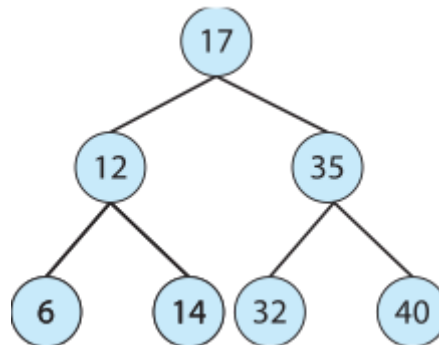
There is nothing to turn in for this assignment, but it will be used for subsequent assignments. It is your responsibility to have the setup ready by the time Nachos assignments are given.

3. Python Programming [25 points]

- Install Python 3.3 or later, if it is not already on your system.
<https://www.python.org/downloads/>
- Use Python *tuple* data structure to represent a *binary search tree*¹ in *preorder* form. For example, the binary search tree in the example below can be represented as
`T = (17, (12, (6, None, None), (14, None, None)), (35, (32, None, None), (40, None, None)))`
- Write a function named `PrintBST(T)` to print such a binary search trees in sorted order ("inorder traversal"). So, if the function is called on the example above, it will print out
6
12
14
17
32

¹ Note: the example in Fig. 1.16 on page 33 of the textbook is not correct. It is not a binary search tree because node 32 should be in the left branch of node 35 and should not be found in the right branch.

35
40



You may assume the tuple-encoded tree is well-formed (i.e., either None or has three elements). `T[0]` is the root, `T[1]` is the left child, and `T[2]` is the right child.

Why tuple encoding? because it is readily readable and usable without having to define any data structure. It lets you focus on the algorithm.

Hints:

- This is an extremely simple assignment. The `PrintBST()` function should take no more than 5-6 lines of Python code. However, as a good programming practice, Python code is usually written with the test case when it is run independently, as shown below.
- For printing to standard-output, use the function syntax of Python 3
`print(n)`
instead of the keyword syntax of Python 2
`print n`
the former syntax is compatible with Python 2 and 3, but the latter is Python 2 only.
- Be consistent with your indentation. In Python, indentation is significant, not just treated as blanks. You should not mix tabs and spaces for indentation.

```
# hw1.py
# your name, student ID
# comments
def PrintBST(T):
    ... # your code here

# begin built-in test case follows your code in the same .py file
# the test case is run when you run this file as top-level,
# but not if it is imported into another Python program as a
module
if __name__ == '__main__':
    T = (17, (12, (6, None, None), (14, None, None)), (35, (32,
None, None), (40, None, None)))
    PrintBST(T)
```