COMP10002 Foundations of Algorithms

Workshop Week 12

Wenbin Cao

October 24, 2019

GitHub Repo: https://github.com/AlanChaw/COMP10002-FoA

Outline

Chapter 13 - Number Representations

- Unsigned types

- Bit operations

- Floating point representations

Chapter 9 - Problem Solving Strategies

Number Representations

Twos-complement representation

Suppose that a machine uses $w = 6$ bits to represent integers. Calculate the twos-complement representations for each of these values: 0, 4, 19, -1, -8, and -31. Verify that $19 - 8 = 11$.

Number Representations

Unsigned types

Type	Range	Format specifier
unsigned char	[0, 255]	%c
unsigned short int	[0, 65,535]	%hu
unsigned int	[0, 4,294,967,295]	%u
...	...	...

Number Representations

Bit operations

Symbol	Operator
&	bitwise AND
|	bitwise inclusive OR
^	bitwise XOR (exclusive OR)
~	bitwise NOT
<<	left shift
>>	right shift

Number Representations

Floating point representations

The floating point types **float** and **double** are stored asï¼š

- a one-bit sign
- a w_e-bit integer exponent
- a w_m-bit mantissa, the leading digit is non-zero

Number Representations

Floating point representations

Number (decimal)	Number (binary)	Exponent (decimal)	Mantissa (binary)	Representation (bits)
0.5	0.1	0	.100000000000	0 000 1000 0000 0000
0.375	0.011	−1	.110000000000	0 111 1100 0000 0000
3.1415	11.001001000011···	2	.110010010000	0 010 1100 1001 0000
−0.1	−0.0001100110011 ···	−3	.110011001100	1 101 1100 1100 1100

Floating point representations

Using the $w_s = 1$, $w_e = 3$ and $w_m = 12$ floating point representations shown in Table 13.4 on page 235, calculate the 16-bit representations for these numbers: 2.0, −2.5, 7.875.

Number (decimal)	Number (binary)	Exponent (decimal)	Mantissa (binary)	Representation (bits)
0.5	0.1	0	.100000000000	0 000 1000 0000 0000
0.375	0.011	−1	.110000000000	0 111 1100 0000 0000
3.1415	11.001001000011···	2	.110010010000	0 010 1100 1001 0000
−0.1	−0.0001100110011···	−3	.110011001100	1 101 1100 1100 1100

Problem Solving Strategies

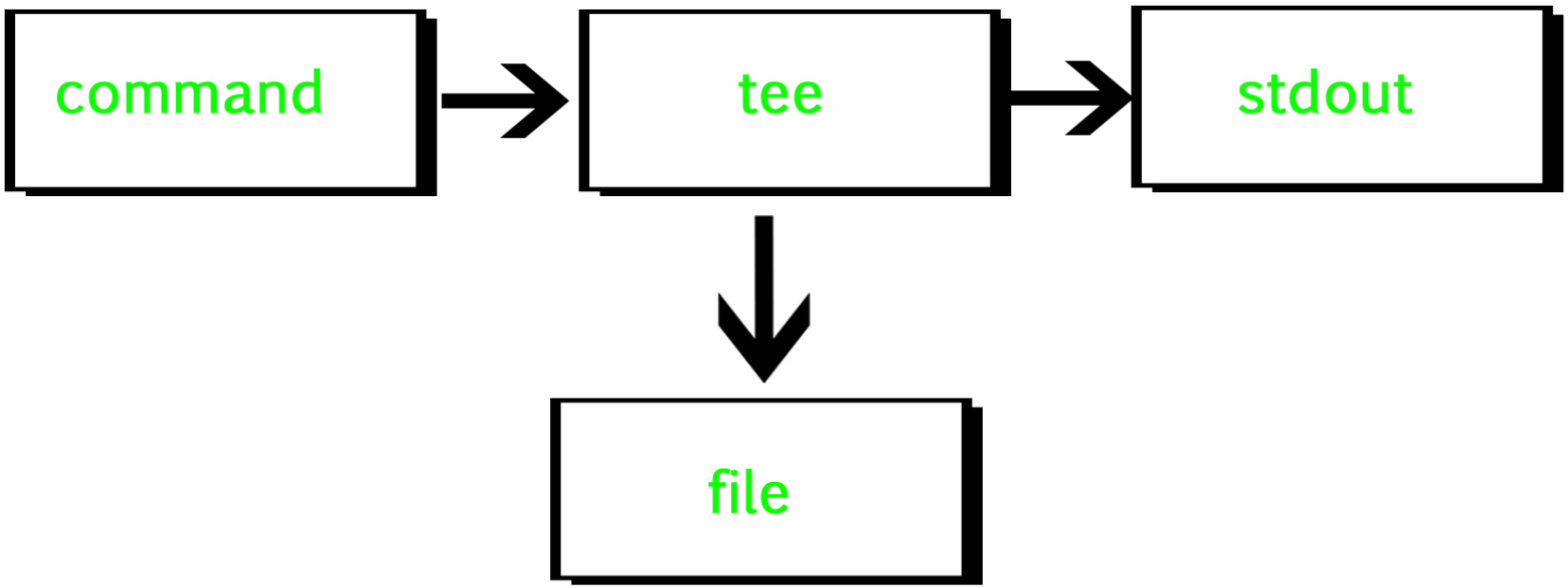
- Generate and test
- Divide and conquer
- Simulation
- Approximation
- Adaptation

Exercise

The Unix `tee` command writes its `stdin` through to `stdout` in the same way that the `cat` command does. But it also creates an additional copy of the file into each of the filenames listed on the command-line when it is executed.

Implement a simple version of this command.

Hint: you will need an array of files all opened for writing.



Tutor feedback

https://apps.eng.unimelb.edu.au/casmas/index.php?r=qoct/subjects