COMP26120 Algorithms and Data Structures
Topic 2: Data Structures

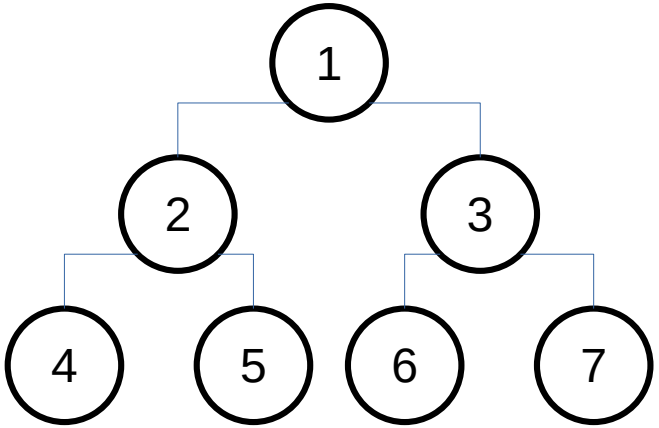
What is a Tree?

Dr. Thomas Carroll
thomas.carroll@manchester.ac.uk
All information on Blackboard

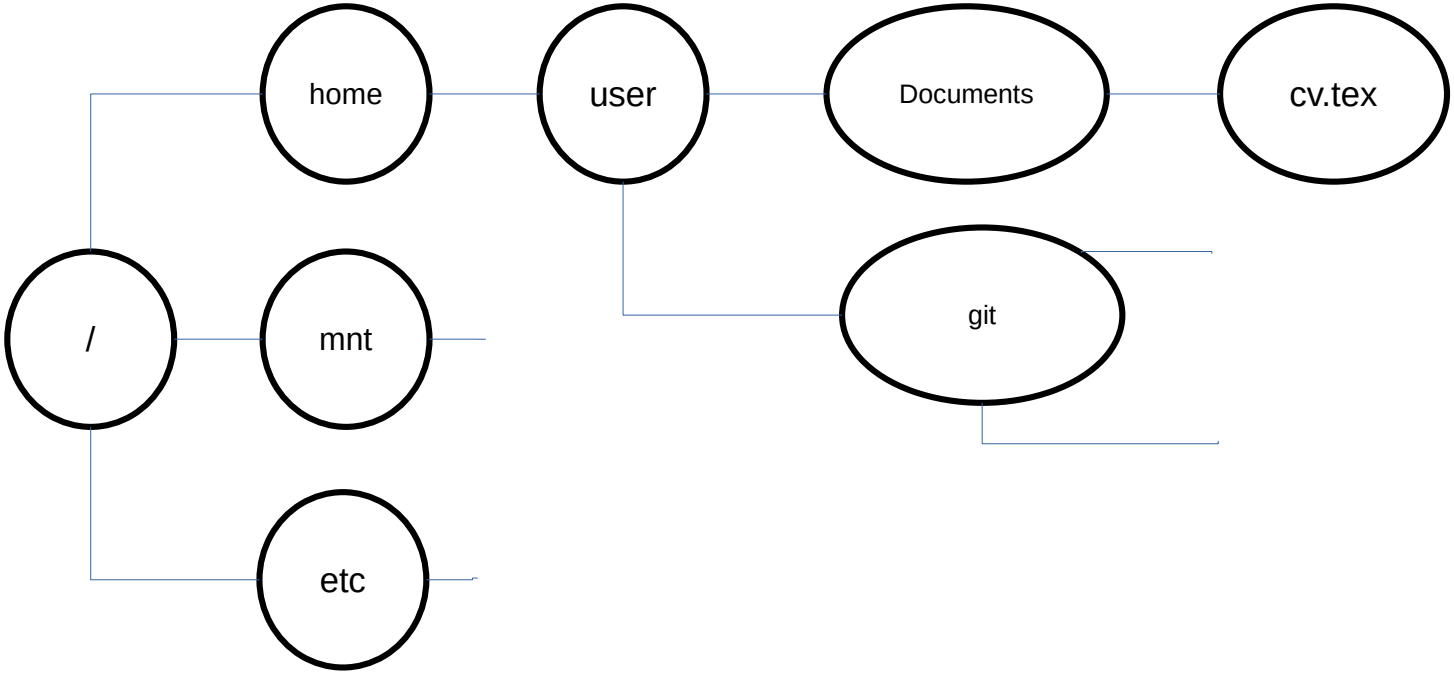
Learning Outcomes

- Understand the concept of a Tree
- Understand the Tree ADT
- Recall examples of Tree usage

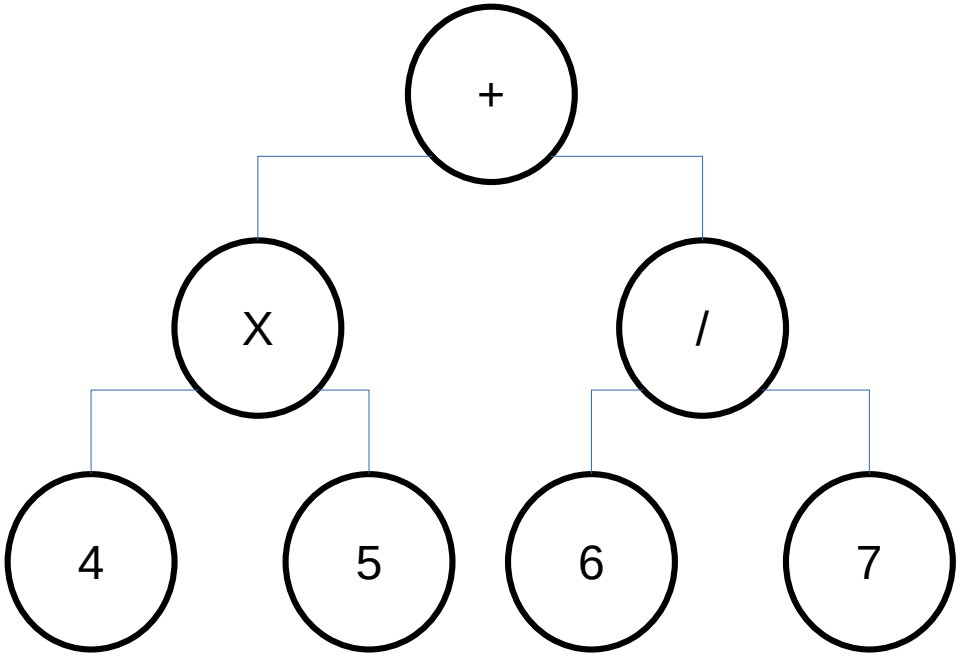
What is a Tree?



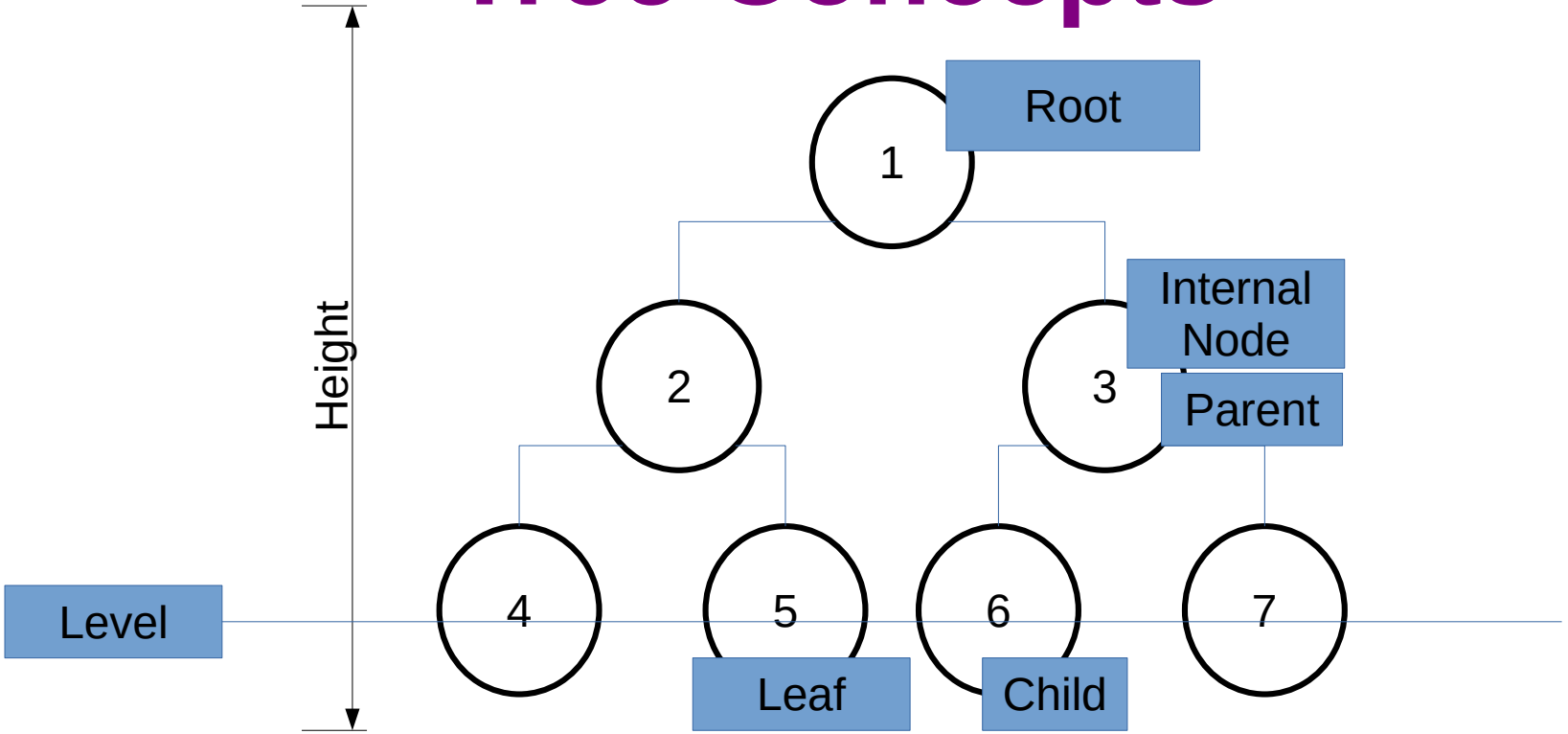
Tree Example: Linux File System



Tree Example: Arithmetic Expression



Tree Concepts



Tree Concepts

- Binary Tree
  - All nodes have at most 2 children
- General Tree
  - At least 1 node has more than 2 children
- Proper Binary Tree
  - Nodes have either 0 or 2 children
- Complete Binary Tree
  - All levels are full, except the last where all nodes are to the left
- Perfect Binary Tree
  - Tree is proper and all external nodes have same depth

Tree ADT

Tree
root(): returns root of tree IsRoot(n): TRUE if n is root node insert(e): Inserts element e into T find(e): returns node storing e (or NULL) remove(n): removes node n from T parent(n): returns the parent node (or NULL) children(n): returns the children nodes of n elements(): returns all elements of T isInternal(n): TRUE if n is internal isExternal(n): TRUE if n is external

Binary Tree ADT

Binary Tree
root(): returns root of tree IsRoot(n): TRUE if n is root node insert(e): Inserts element e into T find(e): returns node storing e (or NULL) remove(n): removes node n from T parent(n): returns the parent node (or NULL) **leftChild(n): returns the left child node of n** **rightChild(n): returns the right child node of n** elements(): returns all elements of T isInternal(n): TRUE if n is internal isExternal(n): TRUE if n is external