Jake Wright

OCR COMPUTING – F453 DEFINITIONS

INTERRUPT

- A message sent to the processor to obtain processor time for a higher priority task
- To avoid delays and loss of data

FILE ALLOCATION TABLE

- A map of where files are stored in the backing store
- Stores filenames and file sizes
- Identifies free space
- Pointers indicate the start of files
- Updated by OS when files are saved, updated or deleted.

WHY IS MEMORY MANAGEMENT NECESSARY?

- To allocate memory to processes to allow separate processes to run simultaneously
- To reallocate memory when necessary
- To deal with memory allocation when paging
- To protect processes from each other
- To allow memory to be shared.

WHY IS VIRTUAL MEMORY NECESSARY?

- To allow programs to run that need more memory than what is available.

HOW IS VIRTUAL MEMORY USED?

- Use of backing store as if it were main memory
- Swap pages between memory and backing store

DISK THRASHING

- Occurs when moving pages between disk and memory
- Disk is relatively slow so high rate of disk access
- More time spent transferring pages than on processing.

WHY IS SCHEDULING NECESSARY?

- Maximise number of users with no apparent delay
- Maximise number of jobs processed as quickly as possible
- Obtain efficient use of processor time
- Ensure no job monopolises processor
- Maximise throughput of CPU.

BOOT FILE

- Supplies Personal Settings

TRANSLATOR

- Converts source code to object code
- Identifies errors in source code.

ASSEMBLER

- Reserves storage for data and instructions
- Replaces mnemonic opcodes by machine codes
- Replaces symbolic addresses by numeric addresses
- Creates symbol table to match labels to addresses
- Checks syntax and offers error diagnostics

INTERMEDIATE CODE

- Improves portability
- Same intermediate code can be obtained from different high level languages.

LEXICAL ANALYSIS

- Source code used as input
- Tokens are created from the reserved words in the program
- A token is a fixed length string of binary digits
- Variable names are loaded into symbol table
- Redundant characters removed e.g. whitespace and comments
- Error diagnostics are given
- Prepares code for syntax analysis

SYNTAX ANALYSIS

- Accepts output from lexical analysis
- Statements are checked against the rules of the language
- Errors reported as a list at end of compilation
- Diagnostics may be given
- If no errors, code is passed to code generator.

CODE GENERATION

- Produces machine code
- Several instructions for each HHL instruction
- Variables and constants are given addresses
- Relative addresses are calculated
- Optimisation: makes code as efficient as possible by reducing number of instructions

LINKERS AND LOADERS

- Combine library routines that are already compiled with program
- Copies modules, from backing store, into memory, ready for execution
- Completes address links to program.

VON NEUMANN ARCHITECTURE

- Single processor manages program control
- Instructions executed sequentially
- Programs and data stored in the same format.

FETCH EXECUTE CYCLE

- PC holds address of next instruction to be executed
- Address copied to MAR
- Increment PC
- Load instruction pointed to by MAR to MDR
- Copy instruction from MDR to CIR
- Decode instruction in CIR

CO-PROCESSOR

- Addition to the main processor for a specific task
- Improves speed
- Example a maths co-processor

ARRAY PROCESSOR

- Processor that allows the same instruction to operate simultaneously on multiple data locations
- Single instruction multiple data

OBJECT

- Instance of a class
- Real world entity

CLASS

- A template for a set of objects that have state and behavior
- Objects can be defined using a class.

ATTRIBUTE

- Information stored about an entity.

GOAL

- A query to be solved.

RELATIONAL DATABASE VS. FLAT FILES

- Avoids data duplication… so less storage wasted
- Less program-data dependencies… so easier to change data format
- Data integrity… only one copy of data stored, less corruption
- More secure/easier to control access to data… views of data can be created.

DATA DESCRIPTION LANGUAGE DDL

- Creates tables, attributes
- Defines data types, primary/foreign keys, validation rules

DATA MANIPULATION LANGUAGE DML

- High level language used to query data
- Used to store or update data.

DATA DICTIONARY

- Names of tables
- Characteristics of data (length, data type)
- Meaning of data columns
- Relationships between tables
- Access rights.

MERGING FILES

- Open existing files
- Create new file
- Check existing files are not empty
- User pointers to identify records for comparison
- Compare records indicated by pointers
- Copy earlier value record to new file
- Move correct pointer
- Repeat
- Until end of file
- Copy remaining records from other file
- Close files
- Assume common key
- Assume if two records are the same, only one is written to the new file.