System Software

A-Level Computer Science · Vocabulary in context

Word	Example
multi-tasking /ˈmʌlti ˈtæskɪŋ/	**Multi-tasking** lets several programs run at once.
paging /ˈpeɪdʒɪŋ/	**Paging** splits memory into blocks of the same size.
spooling /ˈspuːlɪŋ/	**Spooling** keeps print jobs in a queue.
cache /kæʃ/	The **cache** keeps data the processor needs often.
process /ˈprəʊses/	A program that is running is a **process**.
scheduler /ˈʃedjʊlə/	The **scheduler** picks the next process to run.
round robin /raʊnd ˈrɒbɪn/	In **round robin**, each process gets an equal turn.
time slice /taɪm slaɪs/	Each process runs for one **time slice**, then waits.
pre-emptive /priː ˈemptɪv/	A **pre-emptive** scheduler can stop a process at any time.
blocked /blɒkt/	The process is **blocked** while it waits for input.
process control block /ˈprəʊses kənˈtrəʊl blɒk/	The **process control block** stores data about one process.
context switch /ˈkɒntekst swɪtʃ/	A **context switch** saves one process and loads another.
kernel /ˈkɜːnl/	The **kernel** is the core of the operating system.

exercises.lol exercises.com.cn

Word	Example
interrupt handler /ˈɪntərʌpt ˈhændlə/	The **interrupt handler** runs when the interrupt arrives.
interrupt handling /ˈɪntərʌpt ˈhændlɪŋ/	**Interrupt handling** deals with each interrupt in turn.
inter-process communication /ˈɪntə ˈprəʊses kəˌmjuːnɪˈkeɪʃn/	**Inter-process communication** lets processes send data to each other.
pipes /paɪps/	**Pipes** pass data from one process to another.
shared memory /ʃeəd ˈmeməri/	Two processes can use **shared memory** to share data.
virtual address space /ˈvɜːtʃuːəl əˈdres speɪs/	Each process has its own **virtual address space**.
pages /ˈpeɪdʒɪz/	Memory is split into **pages** of the same size.
frames /freɪmz/	RAM is split into **frames** that hold the pages. RAM
page fault /peɪdʒ fɒlt/	A **page fault** happens when the page is not in RAM. RAM
swap file /swɒp faɪl/	The **swap file** holds pages that are not in RAM. RAM
thrashing /ˈθræʃɪŋ/	**Thrashing** happens when pages move in and out all the time.
segmentation /ˌsegmənˈteɪʃn/	**Segmentation** splits memory into blocks of different sizes.
disk thrashing /dɪsk ˈθræʃɪŋ/	**Disk thrashing** makes the computer very slow.
interpreter /ɪnˈtɜːprɪtə/	An **interpreter** runs the program one line at a time.

exercises.lol exercises.com.cn

Word	Example
compiler /kəm'paɪlə/	The **compiler** turns the whole program into machine code.
machine code /mə'ʃi:n kəʊd/	The processor runs **machine code**: only 0s and 1s. 0 1
lexical analysis /'leksɪkl ə'næləsɪs/	**Lexical analysis** removes spaces and comments.
tokens /'təʊkənz/	The code is turned into **tokens**.
syntax analysis (parsing) /'sɪntæks ə'næləsɪs/	**Syntax analysis (parsing)** checks the grammar of the code.
abstract syntax tree /'æbstrækt 'sɪntæks tri:/	The parser builds an **abstract syntax tree**.
syntax error /'sɪntæks 'erə/	A missing bracket is a **syntax error**.
semantic analysis /sə'mæntɪk ə'næləsɪs/	**Semantic analysis** checks that the code makes sense.
code generation /kəʊd ˌdʒenə'reɪʃn/	**Code generation** produces the machine code.
code optimisation /kəʊd ˌɒptɪmaɪ'zeɪʃn/	**Code optimisation** makes the program faster.
symbol table /'sɪmbl 'teɪbl/	The **symbol table** stores each label and its address.
grammar /'græmə/	The **grammar** says which statements are allowed.
Backus-Naur Form /'bækəs nɔ: fɔ:m/	**Backus-Naur Form** writes grammar rules.

exercises.lol exercises.com.cn

Word	Example
production rule /prəˈdʌkʃn ruːl/	Each **production rule** says how to build part of a statement.
terminal /ˈtɜːmɪnl/	A **terminal** is a symbol that cannot be split.
non-terminal /nɒn ˈtɜːmɪnl/	A **non-terminal** is replaced by other symbols.
syntax diagram /ˈsɪntæks ˈdaɪəgræm/	A **syntax diagram** shows the grammar as a picture.
infix /ˈɪnfɪks/	3 + 4 is written in **infix**. 3 + 4
Reverse Polish Notation /rɪˈvɜːs ˈpəʊlɪʃ nəʊˈteɪʃn/	3 4 + is written in **Reverse Polish Notation**. 3 4 +
postfix /ˈpəʊstfɪks/	In **postfix**, the operator comes after the numbers.
stack /stæk/	The last item put on the **stack** is the first one taken off.
precedence /ˈpresɪdəns/	AND has higher **precedence** than OR. AND OR
bytecode /ˈbaɪtkəʊd/	Java code is first turned into **bytecode**. Java

exercises.lol exercises.com.cn