Linguagens:

**o que é uma linguagem de programação?**

R: Uma linguagem de programação é um texto escrito com uma série de instruções, de uma forma especifica, que quando interpretado, ou compilado resulta num programa que faz o que lhe foi instruído.

**que diferenças tem relativamente à linguagem natural (a que**

**falamos)?**

R: É uma linguagem mais técnica, baseada em instruções definidas pelo criador da linguagem de programação.

**qual a diferença entre linguagens compiladas e interpretadas?**

R: Uma linguagem compilada é uma linguagem onde o compilador lê o programa todo e vê se têm erros e compila, enquanto que uma linguagem interpretada, é uma linguagem que é interpretada instrução a instrução.

**em geral quais são mais eficientes?**

R: As mais eficientes são as linguagens compiladas

**Quais são mais adequadas para desenvolvimento rápido de protótipos?**

R: As mais adequadas para desenvolvimento rápido de protótipos são as linguagens intrepertadas

**o que é o código fonte (extensão .c) e o código nativo (a.out)?**

R: O código fonte é o código que o compilador vai ler e a partir dai vai criar o código nativo em binário, que vai ser o código que vai ser executado pelo o processador

**o que é um compilador?**

R: Um compilador, é um programa que traduz instruções para código de máquina que é executado pelo processador

**o que é o assembly de uma arquitectura (extensão .asm)?**

R: Assembly de uma arquitectura é o ficheiro que antecede ao ficheiro binário executável, quando o compilador está a ler um programa cria um código em assembly (.asm) que depois é convertido para código maquina.

**o que é um ficheiro objecto (object file, extensão .o)?**

R: É um ficheiro que liga as bibliotecas todas que o programa necessita para correr

**o que é um ficheiro binário executável? (e.g., a.out)**

R: Um ficheiro executável é um ficheiro em binário, que o processador interpreta e executa

**qual a portabilidade de código fonte e código binário? porquê?**

R: A portabilidade do código fonte é possível, desde que a máquina tenha um compilador ou interpretador adequado, o código binário não tem portabilidade pois foi criado pelo compilador ou interpretador para a arquitetura do processador onde foi criado, logo ao passar para outra máquina pode não ser executado, pois não está compatível com a arquitetura do processador.

**que linguagens são mais portáteis, as compiladas ou as**

**interpretadas? porquê?**

R: As interpretadas, pois não dependem da arquitectura do processador para executar as instruções

**três fases básicas na compilação de um programa para um ficheiro binário executável?**

R: Ler o programa, gerar o código e ligar as bibliotecas necessárias para correr o programa

**o que é o que faz o ligador/linker?**

R: Um linker liga as bibliotecas que o programa necessita para correr o programa

como descobre quais os símbolos em falta num ficheiro objecto?

Primeiro analiza o ficheiro e uso o segmento do simbolo para identificar o código binário em falta, procura por outros símbolos que para os quais se aplique o mesmo código binário, verifica em que parte da memória devem ser colocados esses códigos, e cria o ficheiro.

**o que é e o que faz o carregador/loader?**

R: Um loader é o que vai buscar o programa passar ser executado, e este lê o header para determinar o tamanho e o tamanho do texto e os segmentos de data, cria um espaço na memória para correr o programa, copia os argumentos da linha de comandos para a stack line, inicializa os valores e executa o main.

**o que é um processo?**

R: Um processo é um programa que está a ser executado pelo processador

**o que é um espaço de endereçamento?**

R: Um espaço de endereçamento é o espaço que cada programa que está a ser processado tem direito

**o que é e como está organizado o espaço de endereçamento de um**

**processo?**

R: Está divido pelas fases do loader, header, data, heap, stack, e interrupt

**Circuitos:**

**transistor:**

**o que é?**

R: transístor é um componente eletrónico que controla a passagem de corrente

**como funciona?**

R: Recebe uma corrente e dependendo da informação da corrente, decide se a deixa passar ou se a bloqueia

**tamanho dos transistores actuais (ordem de grandeza)**

14 nm

**implementação das porta lógicas básicas com transistores**

**portas:**

**NOT**

**AND**

**OR**

**NAND**

**Bases**

R: São as portas lógicas

**aplicações das portas lógicas, e.g., SSD/NAND**

**sinais de controlo/relógio**

R: Invertem o sinal do fluxo

**memórias de 1-bit:**

**latch**

R: Circuito que armazena um byte de informação

**flip-flop**

**diferenças**

**registos**

**célula de memória SRAM**

**ALU de 1-bit**

**Adder de 1-bit**

**UNIX/Linux:**

**sistema operativo:**

**o que é?**

R: É o software que suporta as funções básicas do computador

**para que serve?**

R: Serve para organizar o conjunto de programas, aplicações, periféricos que fazem com o que o computador funcione

**quais os mais comuns?**

R: IOS, Windows, Linux, Android

**Quando e por quem foi desenvolvido o sistema Unix?**

R: O sistema Unix foi desenvolvido em 1969, por uma equipa de investigadores da Bell

**quais as características fundamentais de um sistema Unix?**

R: Um sistema hierárquico de ficheiros, execução de ficheiros, uma linha de comandos e algumas funções básicas

**Exemplos de sistemas operativos actuais que são “unixes” e outros**

**que não o são**

R: Linux, Windows

**BIOS:**

**o que é?**

R: Um conjunto de instruções computacionais de firmware que controlam o input e output de operações

**para que serve?**

R: Inicializar o sistema e os I/O devices

**onde podemos encontrá-la?**

R: A BIOS pode ser encontrada numa ROM

**processo de bootstrap:**

**o que é?**

R: É o processo de incializar o sistema operativo

**quais as fases?**

**o que acontece em cada fase?**

BIOS: verifica os I/O, enumera os discos, encontra e carrega o bootloader do disco

Boot Loader: Enumera os OS disponíveis, espera que o utilizador escolha um carrega e executa o OS

Kernel: Inicializa o hardware, o núcleo, decomprime a init e executa a init

**kernel:**

**o que é?**

R: É o núcleo do sistema operativo

**quais as suas funções?**

R: Gestão de processos, gestão da memória, controlo dos dispositivos, gestão dos I/O

**gestão de processos:**

**o que é um processo?**

R: Um processo é um programa que está a ser executado no processador

**um PCB?**

R: Process Control Block

**filas de PCB?**

R: são as filas onde os processos esperam para ser executados

**fila ready e filas associadas a dispositivos?**

R: Fila ready é a fila onde os processos estão a ser executados, filas associadas a dispositivos, são as filas onde os I/O esperam para ser executados

**time quantum?**

R: É o tempo que cada processo tem para executar

**ciclo de execução do processo e operações nas filas?**

**context switch?**

R: Como é controlada a prioridade de processos, e que pode haver a sobreposição de processos

**gestão de memória:**

**diferença entre programa e processo?**

R: Um programa é um conjunto de instruções, um processo é quando essas instruções são executadas

**espaço de endereçamento?**

R: É o espaço que cada processo tem direito para ser executado

**tamanho máximo de espaço de endereçamento com endereços de**

**n bits (2^n bytes, e.g., 32 bits -> 4 GB)**

**swapping**

R:Transferência de informação entre memória e disco

**memória virtual:**

**paginação**

R: a troca de informação entre discos e memória é simples porque as “páginas” têm todas o mesmo tamanho.

**Segmentação**

R: Consiste em dividir a memória em vários segmentos, tal que para cada segmento está associado um processo

**vantagens e desvantagens das duas técnicas**

R: Vantagem da paginação é que como as páginas têm todas o mesmo tamanho, a gestão é simplificada pois é possível, em alguns casos, a intervenção do CPU. Desvantagem da paginação é que paginas que contêm código continuo ou data, podem ser colocados em diferentes regiões da memória o que pode causar um impacto negativo na performance.

Vantagem da segmentação, é que a informação que está relacionada é mantida na mesma região, mas tem como desvantagem o facto de como os segmentos têm diferentes tamanhos, torna se complicado a gestão do espaço disponível na memória

**sistema de ficheiros:**

**unidade básica “ficheiro”, inclui o quê?**

**ficheiros guardados em dispositivos de armazenamento como**

**discos HDD/SSD**

**partição**

**formatação**

**sistema de ficheiros hierarquicos**

**funções de /, /boot, /etc, /home, /proc, /dev, /bin, /**

**sbin, /usr, /lib**

**gestão de dispositivos:**

**dispositivos representados por ficheiros em /dev**

**tipos de dispositivos:**

**character**

**bulk**

**network**

**pseudo**

**formatação de bulk devices com sistemas de ficheiros**

**tipos de sistemas de ficheiros**

**controladores de dispositivos**

**device driver:**

**o que é?**

**para que serve?**

**I/O baseado em interrupts**

**gestão de interrupts por processos**

**DMA**

**Shell:**

**a shell de comandos Unix (Bourne Again Shell): linha de comando e scripts**