Bài Tập môn Kĩ thuật lập trình

Câu 1:Trình bày theo ý nghĩ của riêng "bạn", tại sao bạn học lập trình?

1. Vì kĩ thuật lập trình là một môn học cần thiết và quan trọng trong ngành đang theo học. Nó đóng vai trò là công cụ cốt lõi để hiện thực hiện các ý tưởng điều khiển, giám sát và tối ưu hóa hệ thống.

2. Kĩ thuật lập trình còn là môn học giúp nâng cao và rèn luyện khả năng tư duy, phân tích một vấn đề phức tạp vì trong quá trình thực hiện các dự án sẽ có những lỗi hệ thống và nhu cầu tối ưu hóa thì phương pháp chia nhỏ thành các bước logic để xây dựng giải pháp điều khiển sẽ hiệu quả hơn.

3. Với sự phát triển của xã hội ngày càng hiện đại và càng được số hóa hơn thì: Kĩ thuật lập trình sẽ là một tư duy nền tảng để phát triển nghề nghiệp của bản thân bền vững hơn trong tương lai.

Câu 2:Trình bày cách máy tính thực thi lệnh.

1. Lấy lệnh (Fecth).

a) Mục tiêu: Lấy lệnh tiếp theo từ bộ nhớ chương trình.
b) Hoạt dộng: Bộ đinh vị chương trình (Program Counter - PC) chứa địa chỉ của lệnh cần thực thi. CPU sử dụng địa chỉ này để đọc lệnh từ bộ nhớ và đưa vào bộ nhớ đệm lệnh (instruction cache)

2. Giải mã lệnh (Decode).

a) Mục tiêu: Xác định xem lệnh câng thực hiện thao tác gì và cần dữ liệu ở đâu.
b) Hoạt động: CPU giãi mã lệnh từ bộ nhớ đệm lệnh để hiểu rõ nhiệm vụ cụ thể, chẳng hạn như phép cộng, trừ, hoặc thao tác với bộ nhớ.

3. Thực thi lệnh (Execute).

a) Mục tiêu: Thực hiện lệnh được giải mã.
b) Hoạt động: CPU sử dụng Bộ số học và logic (Arithmetic Logic Unit - ALU) để thực hiện các phép thoán cần thiết . Nếu lệnh yêu cầu, CPU sẽ lấy dữ liệu cần thiết tư thanh ghi (register) hoặc bộ nhớ.

4. Ghi két quả (Write-back).

a) Mục tiêu: Lưu trữ kết quả của lệnh .
b) Hoạt động: Kết quả sau khi xử lí có thể được ghi trở lại thanh ghi hoặc nộ nhớ, tùy thuộc vào yêu cầu.

5. Xử lí ngắt (Interrupt Handling).

a) Mục tiêu: Đáp ứng các sự kiện khẩn cấp từ phần cứng hoặc phần mềm.
b) Hoạt động: Khi một ngắt xảy ra, CPU sẽ tạm dừng việc thực thi lệnh hiện tại để xử lí yêu cầu ngắt, sau đó quay lại thực thi lệnh trước đó.

Câu 3:Trình biên dịch và trình thông dịch là gì? Lập bảng phân loại các ngôn ngữ lập trình theo trình biên dịch và trình thông dịch.

A. Trình biên dịch và trình thông dịch là gì?

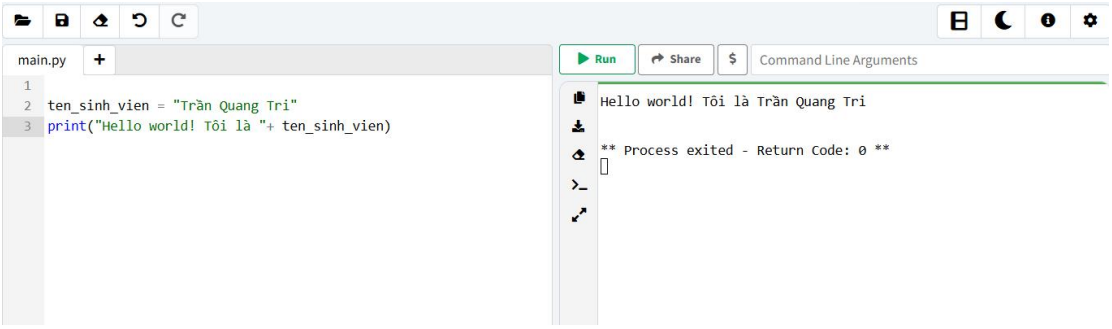
Trình biên dịch (Compiler): Là một chương trình máy tính dịch toàn bộ mã nguồn (chương trình viết bằng ngôn ngữ cấp cao) thành mã máy (ngôn ngữ cấp thấp) hoặc mã đối tượng, sau đó tệp mã đối tượng này mới có thể được thực thi. Quá trình này chỉ diễn ra một lần, giúp chương trình được biên dịch có thể chạy nhanh hơn.

Trình thông dịch (Interpreter): Là một chương trình máy tính đọc mã nguồn và thực thi từng dòng lệnh trực tiếp, mà không cần dịch toàn bộ mã trước. Nó dịch và thực thi từng câu lệnh tuần tự, rồi chuyển sang câu lệnh tiếp theo, làm cho quá trình thực thi chậm hơn so với trình biên dịch.

B. Lập bảng phân loại các ngôn ngữ lập trình theo trình biên dịch và trình thông dịch.

Cơ chế thực thi	Mô tả	Ngôn ngôn ngữ lập trình
Trình biên dịch (Complier)	Dịch toàn bộ mã nguồn sang mã máy trước khi thực thi.	C, C++, Rust, Go, Swift, Pascal, Fortran.
Trình thông dịch (Interpreter)	Dịch và thực thi từng dòng lệnh theo thời gian thực (on-the-fly).	Python, Ruby, JavaScript, PHP, Shell Script.

Câu 4:Viết chương trình in ra màn hình dòng chữ: "Hello world! Tôi là xxx". Trong đó xxx là tên sinh viên


main.py
Run
Share
Command Line Arguments
1
2 ten_sinh_vien = "Trần Quang Tri"
3 print("Hello world! Tôi là "+ ten_sinh_vien)
Hello world! Tôi là Trần Quang Tri
** Process exited - Return Code: 0 **