

1. O Ubuntu é o meu OS principal, então eu abri o terminal com o atalho CTRL + ALT + T.
2. Eu já conhecia os comandos pwd e ls, então não tive surpresa.
3. Eu já conhecia os comandos mkdir (que cria um diretório) e cd (muda de diretório baseado no caminho passado), então não tive surpresa.
4. Pelo o que eu entendi do programa, basicamente ele importa os cabeçalhos stdio.h e o stdlib.h, define a função foo que recebe um parâmetro do tipo float e retorna o valor passado acrescido de 1. Além disso, na função main recebe dois parâmetros que eu não conhecia, mas que depois é explicado que um é a quantidade de itens digitados e o segundo é um array com os itens digitados. Após isso, é definido uma variável "f" do tipo float, em seguida é feito um "If" que verifica se foram passados 2 argumentos. Caso não tenha sido, ele mostra uma mensagem no terminal, o argv[0] eu não entendi e depois ele sai do programa com o exit(0), mas o 0 eu também não entendi. Caso tenha sido, ele atribui a "f" o valor transformado de uma string para um ponteiro float do segundo valor do vetor argv. Após isso, ele printa o valor de argv[1] acrescido de 1 formatado com 2 casas decimais. E então retorna 0.
  - 4.a Não apareceu nenhuma mensagem, ou seja, deu tudo certo na hora de compilar. Ao executar com o comando `./ex1 4.0` foi printado `"5.00"`, o que é esperado dado a função main e foo. Ao executar `./ex1` ele cae na condição em que `argc != 2` e aí ele pega o index 0 do argv (que é o próprio comando, agora eu entendi o argv[0]). Ao executar `"ex1"` ele não encontra o comando, pelo o que o professor disse é porque o caminho para ele não foi encontrado na variável \$PATH
  - 4.b `"gcc -o ex2 labaux.o ex2.o"` esse comando eu não entendi muito bem, mas parece que ele liga os dois objetos no `"ex2"`.
  - 4.c Após compilar e linkar os objetos é executado mostra alguns warnings sobre a falta da declaração da função foo. Em seguida, ao executar `./ex3 4.0` a função retorna 0.00. Eu não entendi muito bem o motivo desse retorno.
5. A função dump recebe um ponteiro p e um inteiro n. Ela cria um ponteiro para unsigned char (que eu acho que é um inteiro) que recebe p. Eu acho que agora p é um ponteiro para o unsigned char. Em seguida é executado um while que tem como condição n ser positivo. Enquanto n for positivo eu vou printando o valor apontado por p1. Após isso eu somo 1 no p1. Nessa parte, como p1 aponta para um lugar p1++ aponta para o próximo endereço. Mas eu também não tenho certeza. Em resumo, parece que a função dump percorre n até 1 e printa os valores apontados por p1 em função de n. No ex4.c temos diversas variáveis criadas e que são passadas como parâmetro da função dump que tem o seu cabeçalho definido. Os parâmetros são o endereço de memória da variável e o tamanho da variável.