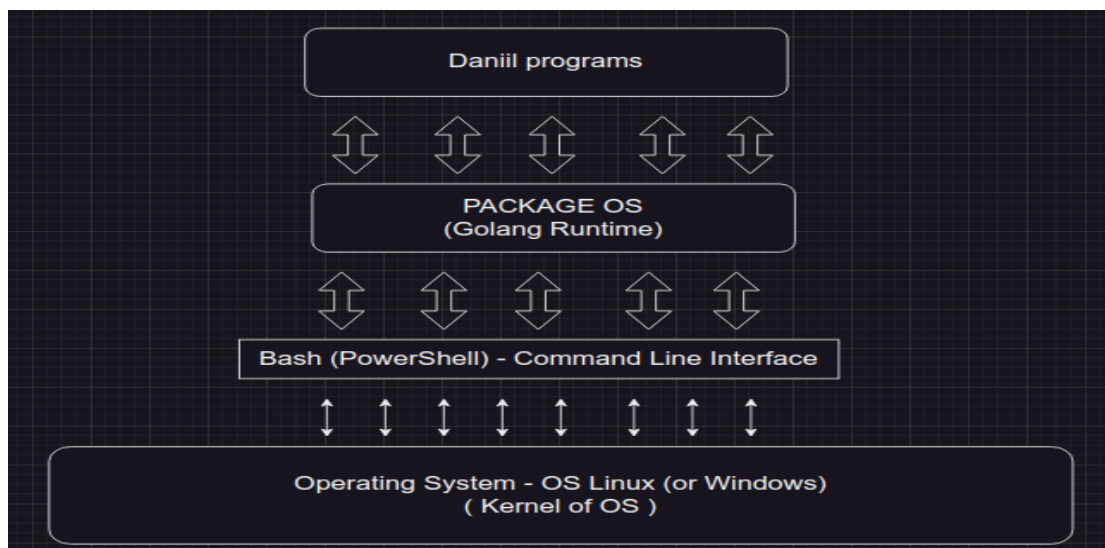


OS (Operating System - Операционная система) - package os

Стандартная библиотека os (package os) - является прослойкой, абстракцией которая повторяет все функции, и операции операционной системы, что бы наша программа могла взаимодействовать с операционной системой при необходимости.



Scr. 1 Description: Как показано в скрине, std. пакет os является некой абстракцией, которая содержит в себе набор функций - методов, структур позволяющих путём их вызова в нашей программы взаимодействовать с операционной системой

The screenshot shows an IDE with two Go files and a terminal. In `main.go`, line 19 calls `args.PrintProgramName()`, which is annotated with a red circle and the number 1. In `07_os.Args.go`, line 8 defines the `PrintProgramName` function, which uses `os.Args[0]` (annotated with a red circle and the number 2) and imports the `os` package (indicated by a red arrow). The terminal at the bottom shows the command `go run cmd/main.go` (annotated with a red circle and the number 3) and its output: `print prog name` followed by `Program name: /tmp/go-build1714876332/b001/exe/main`.

Scr. 2 Description: В данном скрине демонстрируется пример использования package os, точнее os.Args которая является слайсом строк => Args []string -> которая в свою очередь сохраняет все аргументы программы которая его вызывает (имя программы которая его вызывает является его первым элементом т.е. os.Args[0] = "main" - прим. индексы элементов в слайсах, и массивах начинаются с нуля!).

В данной демонстрации программа main печатает свое название путем вызова функции PrintProgramName (1 на скрине), которая внутри своего тела получает его из os.Args[0] стандартной библиотеки os(2 на скрине), и результат выводится в консоль(3).

The screenshot shows an IDE with two files: `main.go` and `args.go`. In `main.go`, line 18 calls `args.PrintProgramName()` (annotated with a red circle 1). In `args.go`, line 5 imports `"os"` (annotated with a red circle 2), and the `PrintProgramName` function (annotated with a red circle 2) uses `os.Args` to retrieve arguments. Lines 13-15 of `args.go` print the program name and arguments (annotated with a red circle 3). The terminal at the bottom (annotated with a red circle 4) shows the command `go run cmd/main.go firstArg secondArg` and the output: `print program name with prog args`, `Program name: /tmp/go-build2094255388/b001/exe/main` (annotated with a red circle 5), `argOne: firstArg` (annotated with a red circle 6), and `argTwo: secondArg`.

Scr. 3 Description: В данной демонстрации программа main печатает свое название, и свои аргументы получения при вызове (firstArg, secondArg) - по сути наша программы большая функция которая вызывается из командной строки, и ей передается входные данные (firstArg, secondArg), а os.Args (2 на скрине) из package os считывает их и сохраняет в себя, начиная с названия программы которая его использовала. Далее наша основная функция (main) вызывает функцию PrintProgramName(1), которая сохраняет внутри своего тела название программы с его аргументами(2). И печатает их(3) -> в свою очередь программы выводить эти данные в консоль(4). В выводе есть название программы(5), и его аргументы(6)