

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**Кафедра технологий программирования**

*Аннотация к дипломной работе*

**Реализация командного интерпретатора для Unix-подобных  
операционных систем**

Лебедев Даниил Петрович

Научный руководитель — доцент кафедры ТП, канд. физ.-мат. наук  
Горячкин В. В.

Минск, 2022

## РЕФЕРАТ

Дипломная работа, 68 страниц, 6 рисунков, 2 таблицы, 5 приложений, 22 источника.

КОМАНДНЫЙ ИНТЕРПРЕТАТОР, UNIX, ПРОЦЕСС, МЕЖПРОЦЕССНОЕ ВЗАИМОДЕЙСТВИЕ, ФОРМАЛЬНЫЙ ЯЗЫК, ГРАММАТИКА, РАЗБОР, ЛЕКСИЧЕСКИЙ АНАЛИЗ, СИНТАКСИЧЕСКИЙ АНАЛИЗ.

*Объект исследования* — командные интерпретаторы как класс программ, а также средства для работы с формальными языками. *Предмет исследования* — проектирование и реализация командного интерпретатора и его входного языка.

*Цели работы* — изучить возможности современных командных интерпретаторов и средства описания формальных языков, а также реализовать собственный интерпретатор и средства разбора его входного языка.

Среди видов пользовательского интерфейса выделяется интерфейс командной строки, также именуемый текстовым. Суть его состоит в том, что пользователь вводит с клавиатуры команды, предписывающие выполнить то или иное действие. После ввода команды исполняются компьютером, а результаты выводятся на экран. Для этого используется специальная программа, называемая командным интерпретатором или оболочкой командной строки. Одно из важнейших мест интерфейс командной строки в целом и командные интерпретаторы в частности занимают в операционных системах семейства Unix.

Командный интерпретатор, разработанный в рамках настоящей дипломной работы, поддерживает основные возможности командных интерпретаторов в Unix-системах — исполнение одиночных команд вместе с перенаправлениями потоков ввода-вывода для них, а также исполнение составных команд: конвейеров, условных и безусловных связок (последовательное или фоновое исполнение входящих в связку команд). В поддержку этой функциональности был разработан входной язык, в который включены соответствующие конструкции. Для этого языка описаны способы его лексического и синтаксического анализа: конечный автомат и формальная грамматика соответственно. На основе спроектированного языка и рассмотренных методов его разбора реализованы лексический и синтаксический анализаторы этого языка, которые впоследствии могут расширяться для поддержки новых конструкций.

## РЭФЕРАТ

Дыпломная работа, 68 старонак, 6 малюнкаў, 2 табліцы, 5 дадаткаў, 22 крыніцы.

КАМАНДНЫ ІНТЭРПРЭТАТАР, UNIX, ПРАЦЭС, МІЖПРАЦЭСНАЕ ЎЗАЕМАДЗЕЯННЕ, ФАРМАЛЬНАЯ МОВА, ГРАМАТЫКА, РАЗБОР, ЛЕКСІЧНЫ АНАЛІЗ, СІНТАКСІЧНЫ АНАЛІЗ.

*Аб'ект даследавання* — камандныя інтэрпрэтатары як клас праграм, а таксама сродкі для працы з фармальнымі мовамі. Прадмет даследавання — праектаванне і рэалізацыя каманднага інтэрпрэтатара і яго ўваходнай мовы.

*Мэты работы* — вывучыць магчымасці сучасных камандных інтэрпрэтатараў і сродкі апісання фармальных моў, а таксама рэалізаваць уласны інтэрпрэтатар і сродкі разбора яго ўваходнай мовы.

Сярод відаў карыстальніцкага інтэрфейсу вылучаецца інтэрфейс каманднага радка, таксама названы тэкставым. Іста яго складаецца ў тым, што карыстальнік уводзіць з клавіятуры каманды, якія прадпісваюць выканаць тое ці іншае дзеянне. Пасля ўводу каманды выконваюцца камп'ютарам, а вынікі выводзяцца на экран. Для гэтага выкарыстоўваецца адмысловая праграма, што завецца камандным інтэрпрэтатарам ці абалонкай каманднага радка. Адно з найважных месцаў інтэрфейс каманднага радка ў цэлым і камандныя інтэрпрэтатары ў прыватнасці займаюць у аперацыйных сістэмах сям'і Unix.

Камандны інтэрпрэтатар, распрацаваны ў рамках гэтай дыпломнай працы, падтрымлівае асноўныя магчымасці камандных інтэрпрэтатараў у Unix-сістэмах — выкананне адзінкавых камандаў разам з перанакіраваннямі струменяў уводу-вываду для іх, а таксама выкананне складовых камандаў: канвеераў, умоўных і безумоўных звязкаў (паслядоўнае ці фонавае выкананне залучаных у звязак камандаў). У падтрымку гэтай функцыйнасці была распрацавана ўваходная мова, у якую ўключаныя адпаведныя канструкцыі. Для гэтай мовы апісаны спосабы яго лексічнага і сінтаксічнага аналізу: канчатковы аўтамат і фармальная граматыка адпаведна. На аснове спраектаванай мовы і разгледжаных метадаў яго разбору рэалізаваны лексічны і сінтаксічны аналізатары гэтай мовы, якія пасля могуць пашырацца для падтрымкі новых канструкцый.

## ABSTRACT

Thesis, 68 pages, 6 pictures, 2 tables, 5 appendices, 22 sources.

COMMAND-LINE SHELL, UNIX, PROCESS, INTERPROCESS COMMUNICATION, FORMAL LANGUAGE, GRAMMAR, PARSING, LEXICAL ANALYSIS, SYNTACTIC ANALYSIS.

The *object of research* is command-line shells as a class of programs and means of working with formal languages. The subject of research is projecting and implementing a command-line shell and its input language.

The *purpose* is to study the features of modern command-line shells and means of describing formal languages as well as to implement our own shell and means of parsing of its input language.

Among the types of user interface, the command-line interface, also referred to as text interface, stands out. Its essence consists in the fact that the user inputs commands from the keyboard prescribing to perform some action. After inputting, the commands are executed by the computer, and the results are displayed on the screen. To do this, a special program is used, called a command interpreter or a command-line shell. One of the most important places is occupied by the command line interface in general and command-line shells in particular in Unix operating systems.

The command-line shell developed within the framework of this thesis supports the main capabilities of command-line shells in Unix systems: the execution of single commands together with redirections of I/O streams for them, as well as the execution of composite commands such as pipelines, conditional and unconditional bundles (sequential or background execution of commands included in the bundle). In support of this functionality, an input language has been developed, which includes the corresponding constructs. For this language, the methods of its lexical and syntactic analysis are described; these are a finite automaton and formal grammar respectively. On the basis of the designed language and the considered methods of its analysis lexical and syntactic analyzers of this language are implemented; they can subsequently be expanded to support new constructions.