Lab 1 Introduction

Operating Systems Course

EDA093/DIT401

Logistics

- You **need** to be in a group to have access to the lab.
- **2 people** per group.
- You need to pass the lab 1 preparation test before you can submit!

Shell: A Quick Recap

- A *Command Line Interpreter* that provides a user interface to the operating system.
- Basic tasks
  - Get input commands from the user.
  - Execute commands and display output.
- The shell itself does not understand commands (with few exceptions)!
- It only searches for the binary of the given command and executes it with provided arguments.

Your task

- Develop a basic shell program, called **lsh**.
- **lsh** should be able to replicate the functionality of UNIX shells like sh, bash, csh, etc.

Prerequisites

- Operating System Concepts
  - Parent and child processes
  - Zombie/defunct processes
  - Background processes
  - UNIX signals, signal handling
  - System calls, such as fork, exec, execvp, clone, etc.
- Basic familiarity with Linux

Lab 1 Specifications 1

- Allow users to enter commands to execute programs installed on the system.
- Execute any binary found in the PATH environment variable.
- Example 1: Commands **without** arguments
  - ls, date, ps, etc.
- Example 2: Commands **with** arguments
  - ls –l, date –R, ps aux, etc.

Lab 1 Specifications 2

- Execute commands in the **background**
- For example: sleep 20 &
  - The & sign will spawn the sleep process in the background
  - lsh will be ready to immediately take the next command from the user.

Lab 1 Specifications 3

- Support the use of one or more **pipes**.
- Example: ls | sort | wc -w
  - ls outputs the list of all the files and directories in the current directory.
  - sort reads the output of ls and sorts it
  - wc reads the output of sort and counts the number of words it contains.
  - ls, sort and wc communicate using pipes

Lab 1 Specifications 4

- Allow **redirection** of stdin and stoud to files
- Example: wc –l < /etc/passwd > out.txt
  - The command creates a new file out.txt which contains the number of user accounts in the machine.

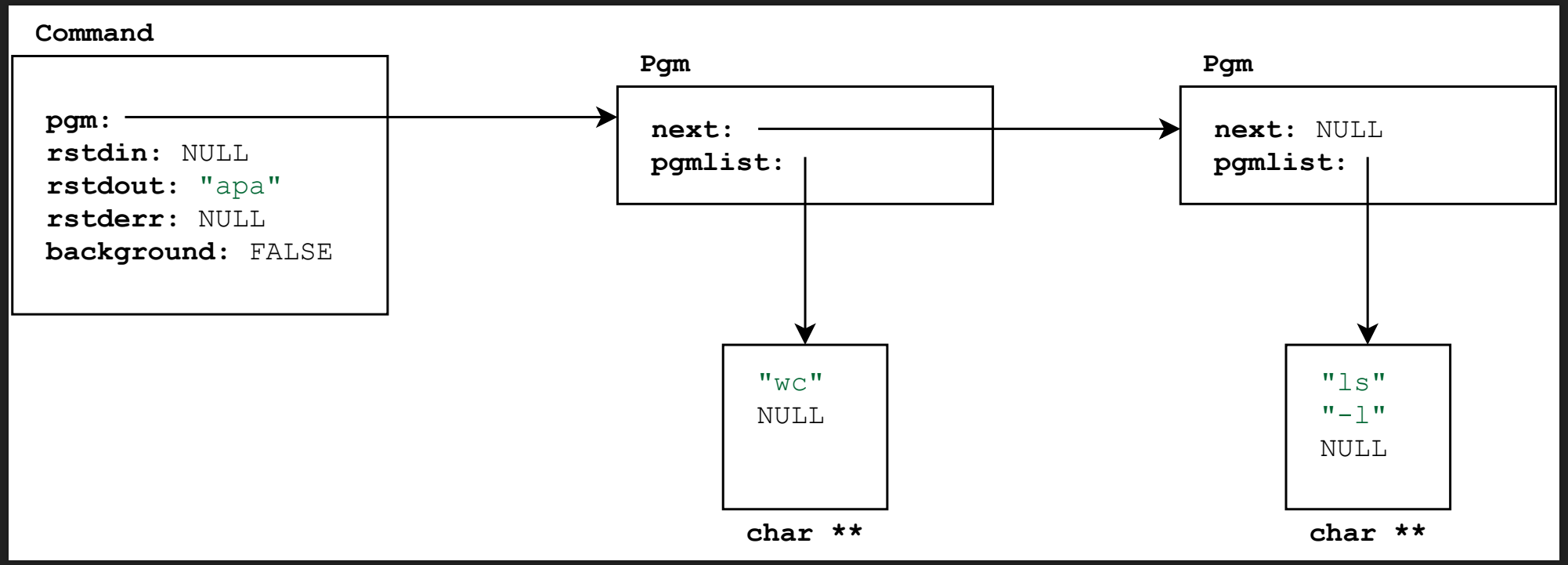
Lab 1 Specifications 5 & 6

- cd and exit are provided as built-in functions
- Pressing Ctrl-C should terminate the execution of a foreground program running on your shell but **not** the execution of the shelf itself
- Ctrl-C should **not** terminate any background jobs either.

Getting Started

- Finish the **preparation test** in canvas.
- Download the from **lab instructions** and the **skeleton code** from canvas.
  - You must base your solution on the skeleton code.
  - It includes a parser which converts a command string to a command data structure.
  - Prints the command entered by the user.

Parsing Commands



Testing

- **Implement** and test the specifications one at a time and in the order given.
- **Verify** using self-test examples listed in lab instructions (and include in your report!).
- If you are unsure of the correct behavior, **compare** with bash or similar shell.
- Your shell should always remain in a **usable state**!
- Make sure your shell does not create **zombies**!

Before Submission

- **Clean Code!**
  - Indentation (most editors can do this automatically!)
  - Meaningful variable names.
  - Explain difficult parts with comments.
  - No debugging code (printfs etc.)
- **Test correctly!**
  - Remove any temporary files before testing.
  - make clean and make again before testing.
  - We will test your code on remote11.chalmers.se. **You must do the same!**
    Submissions that do not compile / work on remote11.chalmers.se will fail!
  - Use the prepare-submission script to prepare your files and make sure you fulfil the requirements.

Report & Attendance

- You need to submit a lab report.
  - Provide a high-level description of the implementation of each feature.
  - Discuss problems you encountered and how you solved them.
  - Describe the execution of self-test examples (including the output of your shell!) and answer the related questions.
- You need to attend at least one lab session.
  - Discuss your solution with the TAs.

Read the lab instructions carefully!