**Pogrebnak Illua**

1) The main functions of the task scheduler in the OS:

1. Assignment of resources:

● Description: Allocation of processor time and other system resources between different tasks.

● Windows: Uses a multitasking scheduler to use resources efficiently.

● Linux: Implements kernel-level scheduling, such as Completely Fair Scheduler (CFS).

2. Setting priorities:

● Description: Determining the importance and urgency of tasks.

● Windows: Has a priority system to manage the importance of tasks.

● Linux: Uses "nice" to prioritize processes.

3. Distribution of resources:

● Description: Optimal distribution of processor time and other resources between tasks.

● Windows: Uses a scheduler that regulates the frequency of processor calls.

● Linux: Has allocation algorithms at the kernel level, such as Completely Fair Queuing (CFQ) for I/O.

4. I/O planning:

● Description: Organization of I/O to avoid delays due to blocking operations.

● Windows: Uses optimization techniques to better manage I/O.

● Linux: Implements I/O scheduling algorithms such as Completely Fair Queuing (CFQ). \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comparison of task scheduling in Windows and Linux: Windows:

Multilinearity:

Windows uses a multitasking scheduler that can manage many processes simultaneously. Priorities: Users can prioritize tasks through the task manager. Task Scheduler: Windows has a built-in task scheduler to automate various processes.

Linux:

Control via Nice and Renice: The priority of processes is determined by the value "nice". A higher nice value indicates a lower priority.

Cron: Used to periodically run tasks on a specified schedule. Kernel-level scheduling: Kernel-level tasks can be scheduled using Completely Fair Scheduler (CFS) or Completely Fair Queuing (CFQ) principles.

Cron scheduler in Linux OS:

1. Basic principles of work:

● Cron: This is a service in Linux that allows users to schedule tasks or commands to run at a specific time or periodically.

● Cron tables: Tasks to be executed are defined in cron tables that have an execution schedule.

2. The structure of the Cron table:

● Each user can have his own cron table.

● Cron tables are located in the /etc/cron.d/ directory or can be configured using the crontab command.

3. Syntax of the cron-table entry:

● The record has the following format: minute hour day month day\_of\_week command.

● Allows you to specify the time (minutes, hours), day of the month, month and day of the week. 4. Examples of records:

● \* \* \* \* \* command: Execute command every minute. ● 0 2 \* \* \* command: Execute the command at 2 o'clock every day.

● 0 1 \* 5 \* command: Execute the command at 1 o'clock every day in May.

5. Commands for working with crontab:

● crontab -l: View the current cron table.

● crontab -e: Edit or create a new cron table.

● crontab -r: Delete the current cron table. 6. Alternatives:

● Anacron: Used to perform tasks on behalf of the user even if the computer is turned off at the specified time. An alternative to cron for those cases when the computer is not always on.

● Systemd Timers: The systemd init daemon also provides options for scheduling tasks via timers.

● Alternative implementations of cron:

There are different implementations of cron, such as dcron, fcron, which may have some extensions or differences in functionality.

**Tseluiko Stanislav**

2.

3. To install the Anacron task scheduler on Linux, you need:

• Open the terminal.

• Update package information using the sudo apt update command

• Install Anacron: sudo apt install anacron

• Check that Anacron is installed correctly: anacron –version After installing Anacron, you can edit the configuration file to add your tasks for scheduling. This file contains the rules for running tasks in the Anacron system. You can use any text editor in the terminal (eg nano, vim or gedit) to edit this file.

**Translated by Barabash Matvii**