# Environment Variables in UNIX

| | |
|---|---|
| **Name** | : Amitendu Bikash Dhusiya |
| **Roll No.** | : 15242723010 |
| **Stream** | : BCA |
| **Semester** | : 6th |
| **Paper Name** | : Unix & Shell Programming |
| **Paper Code** | : BCAC602 |

# Concept of the Environment

- **Definition**: A collection of "*key-value*" pairs passed to a process that defines the setting in which the process runs.
- **Process Inheritance**: When a process is created, it inherits a copy of its parent's environment.
- **Storage**: Environment variables are stored in a specific area of the process's memory (the stack).

# Shell vs. Environment Variables

- **Local (*Shell*) Variables**: Defined within a shell but not passed to child processes.
    - **Example**: `NAME="Gemini"`
- **Environment Variables**: Defined and "exported" so that any sub-shell or command can see them.
    - **Example**: `export NAME="Gemini"`

# The `export` Command

- **Mechanism**: The export command marks a shell variable for inclusion in the environment of subsequently executed commands.

- **Syntax**:
  - `VAR=value; export VAR` (*Traditional*)
  - `export VAR=value` (*Modern Bash/zsh*)

- **Check Export Status**: Use *`declare -p VAR`* to see if the "*export*" attribute (*`-x`*) is set.

# Key Global Variables

- `PATH`: The search path for commands. A colon-separated list of directories.
- `HOME`: The **current user's home directory** (referenced by ~).
- `PS1`: Defines the appearance of the **primary shell** prompt.
- `IFS`: **Internal Field Separator**—defines how the shell splits words (default is space, tab, newline).

# The PATH Variable in Depth

- **The Search Logic**: When a command is entered without a path (e.g., `gcc`), the shell scans `PATH` from left to right.
- **Security Note**: It is a security risk to include the current directory (`.`) at the start of your `PATH`.
- **Modification**: `PATH=$PATH:/opt/bin` (Adds to the end)
  - `PATH=/opt/bin:$PATH` (Adds to the beginning/takes priority)

# Managing Variables with Commands

- `env`: Runs a program in a modified environment or lists exported variables.
- `unset`: Removes a variable from both the shell and the environment.
- `readonly`: Marks a variable so it cannot be changed or unset during the session.
- `printenv`: Specifically used to print environment variables (*ignoring local shell variables*).

# Environment Persistence & Startup Files

- Variables defined in the terminal are lost on logout. For persistence, they are added to startup scripts:
  - **Login Shells:** `.profile` or `.bash_profile` (*Read once at login*).
  - **Non-Login Shells:** `.bashrc` or `.zshrc` (*Read every time a new terminal window opens*).
  - **Global Configuration:** `/etc/profile` affects all users on the system.

# Summary & Key Takeaways

- **Core Purpose:** Environment variables act as global configuration settings that define the behavior of the Unix shell and its child processes.
- **The Inheritance Rule:** Variables move **downward**, not upward. A parent process (***shell***) passes its exported variables to child processes, but a child cannot change the parent's environment.
- **Key Commands to Remember:**
  - `export`: Promotes a local shell variable to an environment variable.
  - `unset`: Deletes a variable from the current session.
  - `printenv / env`: Displays the current environment state.
- **Vital Variables:**
  - `PATH`: The most critical variable; determines where the shell looks for executables.
  - `HOME`, `PWD`, `USER`: Define the user's context and location.
  - `PS1`: Controls the shell's visual interface (the prompt).

# Thank You