Local and Global Unix Variables

By Dhandapani Yedappalli Krishnamurthi Sep 30, 2025

🔹 1. Subprocess in Shell

- In Unix/Linux, whenever you run a **command in a script or inside `$(...)` or a pipeline**, a **subprocess (child shell)** is created.
- A **subprocess inherits variables from the parent shell** but **does not pass new variables back to the parent**unless explicitly exported.

🔹 2. Local Variables

- By default, variables you define in a shell are **local to that shell** (not available to subprocesses).

```
#!/bin/bash
name="Kapil"
echo "In parent shell: $name"

# Run a subprocess
```

```
bash -c 'echo "In child shell: $name"'
```

✅ **Output:**

```
In parent shell: Kapil
In child shell:
```

👉 The variable `name` is **not visible in the subprocess**.

🔹 3. Using `export`

- If you want a variable to be **available in subprocesses**, you must **export** it.

```
#!/bin/bash
name="Tharun"
export name   # Make it available to child processes

bash -c 'echo "In child shell: $name"'
```

✅ **Output:**

```
In child shell: Tharun
```

👉 Now the subprocess can see `name` because it was exported.

🔹 4. Example: Local vs Exported Variables

```
#!/bin/bash
# Local variable (not exported)
city="Chennai"

# Exported variable
state="Tamil Nadu"
export state

echo "Parent shell -> city: $city, state: $state"

bash -c 'echo "Child shell -> city: $city, state: $state"'
```

✅ **Output:**

```
Parent shell -> city: Chennai, state: Tamil Nadu
Child shell -> city: , state: Tamil Nadu
```

👉 Only `state` is available in the child shell, `city` disappears.

🔹 5. Subprocess Example with Function + Export

```
#!/bin/bash
# Function creates a subprocess
function my_function {
    local local_var="Sneha"    # local to function
    global_var="Sangeetha"     # available in script
    export exported_var="Kapil"  # available to subprocess
}

my_function

echo "Script sees: local_var=$local_var, global_var=$global_var, exported_var=$exported_var"

bash -c 'echo "Subprocess sees: local_var=$local_var, global_var=$global_var, exported_var=$exported_var"'
```

✅ **Output:**

```
Script sees: local_var=, global_var=Sangeetha,
exported_var=Kapil
Subprocess sees: local_var=, global_var=,
exported_var=Kapil
```

👉 Key things:

- `local_var` is only visible **inside the function**.
- `global_var` is visible in the script but **not exported** → child shell can't see it.
- `exported_var` is visible everywhere → script and subprocess.

🎯 Key Takeaways

1. **Local variable** → only inside function or current shell.
2. **Global variable** → available in the script but **not in subprocesses**.
3. **Exported variable** → visible to subprocesses.
4. **Subprocess** → child shell created by `bash -c`, pipelines, command substitution, etc.
