5-06-2020

Today we are going to learn communicating with Processes.

According to world wide updates — according to our need, Java 9 has one very necessary update known as — **JPMS** Java Process Management System

By using JPMS feature we can feel very confortable to communicate with running processes of any operating system. This feature is specially very useful while developing device drivers.

In java 9, it is one of the best feature — known as — **Process API updates**

In earlier version, it was very difficult to communicate with processor since we have to write native codes there.

Some tasks related to process we can perform easily are —

① Process ID of current running process.

② Create our own new process.

③ We can destroy any already running process.

④ We can know a complete process information about any running process — eg. who is using this process, which are child process, parent process etc.

⑤ Child process information of any process.

⑥ Parent process information of any process.

In other words, until Java 8, communicating with processor is very difficult, we required to write very complex native code and we have to use 3rd party jar files.

The way of communicating with processor is varied from system to system (i.e., OS. to OS.) For example, in windows one way, but in Mac other way. Being a programmer we have to write code based on operating system, which makes programming very complex.

To resolve this complexity, JDK 9 Engineers introduced several enhancements to **Process API**. By using this updates, we can write java code to communicate with any processor very easily. According to world wide java developers, Process API updates is the number 1 feature in Java 9.

With this enhanced API, we can perform the following activities very easily —

1. Get the PID (Process ID) of running process.
2. Create a new process.
3. Destroy already running process.
4. Get the process handles for processes.
5. Get the parent and child processes of running process.

6. Get the process information like owner, children, etc.

Java 9 Process API updates :-

existing
- Process class — java.lang package is already there in old versions. new methods — pid(), info() ... added
- ProcessBuilder class — java.lang package is also present.

new
- ProcessHandle interface — is new in Java 1.9.
- ProcessHandle.Info (sub-interface) — is new in Java 1.9

All API's are inside java.lang package hence, we no need to import any package.

New Features of Java 9 Process API :-

1. Added several new methods (like pid(), info() etc.) to Process class.
2. Added several new methods (like startPipeline()) to ProcessBuilder class. We can build operating system process by using ProcessBuilder class.
3. Introduced a new powerful interface ProcessHandle. With this interface, we can access current running process, We can access parent and child processes of a particular process etc.
4. Introduced a new interface ProcessHandle.Info, by using this we can get complete information of a particular process.

Now, question is — how to get ProcessHandle object ?

① To get ProcessHandle of current Running Process —

```
ProcessHandle handle = ProcessHandle.current();
```

② To get the ProcessHandle of the given Process object —

```
ProcessHandle handle = p.toHandle();
```

③ To get ProcessHandle of the specified process ID :—

```
ProcessHandle handle = ProcessHandle.of(PID);
```

Here the return type is Optional, because PID may exist or not. Hence use is like —

```
Optional<ProcessHandle> opt = ProcessHandle.of(PID);
ProcessHandle handle = opt.get();
```

Example ① :- To get Process ID of current Process.

```
public class PidDemo{
public static void main(String [] args) throws Exception
  {
ProcessHandle handle = ProcessHandle.current();
long pid1= handle.pid();
System.out.println("The PID of current JVM instance:"
                                    + pid1);
  Thread.sleep(1000000);
  }
                                    }
```

ProcessHandle.Info and its methods.

To get complete information of a running process or particular process.

```
ProcessHandle handle = ProcessHandle.current();
ProcessHandle.Info info = handle.info();
```

Methods of ProcessHandle.Info interface:-

① user()

```
Optional<String> opt = info.user();
System.out.println("User:" + opt.get());
```

② command()

```
Optional<String> opt = info.command();
System.out.println("Command:" + opt.get());
```

③ startInstant()

```
Optional<Instant> opt = info.startInstant();
System.out.println("starting time:" + opt.get());
```

④ totalCpuDuration()

```
Optional<Duration> opt = info.totalCpuDuration();
System.out.println("Total CPU time:" + opt.get());
```

Explanation — Demo:- To get complete information of a current running process:—

```
public class Test
{
    public static void main(String [] args) throws
                                                Exception
    {
```

```
ProcessHandle handle = ProcessHandle.current();
ProcessHandle.Info info = handle.info();
System.out.println("Complete Process Info:" + info);
System.out.println("User :" + info.user().get());
System.out.println("Command :" + info.command().get());
System.out.println("Starting Time :" + info.startInstant().
                                          get());
System.out.println("Total CPU duration used :" +
                        info.totalCpuDuration().get());
    }
}
```

Once we get ProcessHandle.Info information, then we get complete process information by info

If we want to print each field separately-then we have several methods on which we can call get() to print optional objects to convert into ProcessHandle object, such as:-

- info.user().get()
- info.command().get()
- info.startInstant().get()
- info.totalCpuDuration().get()

— Thats all today —