

[Home](#) > [Tools](#) > ♥ 14 Unix interview questions & answers for Java developers

# ♥ 14 Unix interview questions & answers for Java developers

Posted on [October 3, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — [No Comments](#)



**Q1** How do you remove the Control-M characters from a file?

**A1** Using the sed command that replaces Control-M with nothing

```
1 sed 's/^M//g' ReadWriteFile.sh > ReadWriteFileNew
```

**Note:** The ^M is typed on the command line with ctrl+v and ctrl+M

```
1 dos2unix ReadWriteFile.sh ReadWriteFile.sh
```

**Q2** How will you search for a property named “inbox” within a number of .properties files including all sub folders?

**A2** Using the find and grep commands

**600+ Full Stack Java Interview Questions Answered**  
♥Free ♦FAQs [Mouse Hover for full title text]

[open all](#) | [close all](#)

☒ [Ice Breaker Interview](#)

☒ [Core Java Interview C](#)

☒ [JEE Interview Q&A \(3](#)

☒ [Pressed for time? Jav](#)

☒ [Job Interview Ice B](#)

☒ [FAQ Core Java Jok](#)

☒ [FAQ JEE Job Inter](#)

♦ [12 FAQ JDBC](#)

♦ [16 FAQ JMS ir](#)

♦ [8 Java EE \(aka](#)

♦ [Q01-Q28: Top](#)

♦ [Q29-Q53: Top](#)

01: ♦ [12 Web ba](#)

06: ♦ [MVC0, MV](#)

[JavaScript mista](#)

[JavaScript Vs Ja](#)

[JNDI and LDAP](#)

[JSF interview Q](#)

[JSON interview](#)

```
1 $find . -type file -name "*.properties" | xargs g
2 $find . -type f -name "*.properties" -exec grep "
```

**Q3** How will you run a shell script even after you log out of the session?

**A3** “nohup” to run even after logging out of the session. & to run as a background process. The output will be directed to nohup.out

```
1 $nohup ./myapp.sh &
```

If you don't want the output to be directed to nohup.out.

```
1 $nohup ./myapp.sh > /dev/null 2>&1 &
```

**Q4** How will identify the jar file that has a particular class or resource file?

**A4** For example, to find the jar files that has the “MyConnection.class” class file. This is handy for identifying class loading issues in Java.

```
1 find . -name '*.jar' -print0 | xargs -0 -I '{}' "
```

**Q5** How will reuse some of the commands that you have already used?

**A5** Using the “history” command.

```
1 $history # list the history
2 $history | grep mkdir # list mkdir comman
3 !$cd # execute last cd c
4 !$35 # execute command n
5
```

**Q6** How will find the list of log files that have userid “B1234”?

**A6** Using the “grep” command with option “l”.

```
1 grep -l "B1234" *.log
```

- [FAQ Java Web Ser](#)
- [Java Application Ar](#)
- [Hibernate Job Inter](#)
- [Spring Job Interview](#)
- [Java Key Area Ess](#)
- [OOP & FP Essenti](#)
- [Code Quality Job I](#)
- [SQL, XML, UML, JSC](#)
- [Hadoop & BigData Int](#)
- [Java Architecture Inte](#)
- [Scala Interview Q&As](#)
- [Spring, Hibernate, & I](#)
- [Testing & Profiling/Sa](#)
- [Other Interview Q&A I](#)
- [Finance Domain In](#)
- [FIX Interview Q&A](#)
- [Groovy Interview C](#)
- [JavaScript Interview](#)
- [Unix Interview Q&A](#)
  - ♥ 14 Unix intervi
  - ♥ Hidden Unix, C
  - sed and awk to v
  - Shell script inter
  - Unix history com
  - Unix remoting in
  - Unix Sed comm
- [Free Java Interview](#)

## 16 Technical Key Areas

[open all](#) | [close all](#)

- [Best Practice \(6\)](#)
- [Coding \(26\)](#)
- [Concurrency \(6\)](#)
- [Design Concepts \(7\)](#)
- [Design Patterns \(11\)](#)
- [Exception Handling \(3\)](#)
- [Java Debugging \(21\)](#)
- [Judging Experience I](#)

1. Use the up arrow to view the previous command and press enter to execute it.
2. Type !! and press enter from the command line to execute the last command again
3. Type !-1 and press enter from the command line to execute the last command !-2 for command before the last and so on.
4. Type history followed by enter, which prints a list of last few commands, and then !25 to execute a particular command

**Q7** What do you understand by 2>&1 in UNIX?

**A7** In UNIX you have STDIN, which is denoted by number 0, STDOUT, which is denoted by 1, and STDERR, which is denoted by 2. So, the above line means, the error messages go to STDERR, which is redirected to STDOUT. So, the error messages go to where ever the STDOUT goes to. For example, The following command creates a multiple directories for a maven based Java project. if the the directory creation is successful, change directory to “project” and print the directory tree structure. The STDOUT and STDERR are directed to the file named maven-project.log under the project folder.

```
1 mkdir -p project/{src/{main/{java,resources},test
```

The output will be something like

```
1 ./src
2 ./src/main
3 ./src/main/java
4 ./src/main/resources
5 ./src/test
6 ./src/test/java
7 ./src/test/resources
```

**Q8** What is /dev/null?

**A8** It is a blackhole.

```
1 $ cd temp > /dev/null 2>&1 && echo $(pwd)
```

will fail silently and nothing will be printed if there is no “temp” folder. The message has gone into the blackhole. If there is a

- ⊞ [Low Latency \(7\)](#)
- ⊞ [Memory Management \(13\)](#)
- ⊞ [Performance \(13\)](#)
- ⊞ [QoS \(8\)](#)
- ⊞ [Scalability \(4\)](#)
- ⊞ [SDLC \(6\)](#)
- ⊞ [Security \(13\)](#)
- ⊞ [Transaction Management \(13\)](#)

## 80+ step by step Java Tutorials

[open all](#) | [close all](#)

- ⊞ [Setting up Tutorial \(6\)](#)
- ⊞ [Tutorial - Diagnosis \(2\)](#)
- ⊞ [Akka Tutorial \(9\)](#)
- ⊞ [Core Java Tutorials \(2\)](#)
- ⊞ [Hadoop & Spark Tutorial \(1\)](#)
- ⊞ [JEE Tutorials \(19\)](#)
- ⊞ [Scala Tutorials \(1\)](#)
- ⊞ [Spring & Hibernate Tutorial \(1\)](#)
- ⊞ [Tools Tutorials \(19\)](#)
- ⊞ [Other Tutorials \(45\)](#)
  - ⊞ [SQL Tutorial \(1\)](#)
  - ⊞ [Drools Tutorial \(5\)](#)
  - ⊞ [Apache POI Tutorial \(1\)](#)
  - ⊞ [DOS Tutorial \(2\)](#)
  - ⊞ [Dozer Tutorial \(2\)](#)
  - ⊞ [ESB Tutorials \(1\)](#)
  - ⊞ [Groovy Beginner Tutorial \(1\)](#)
  - ⊞ [JasperReport Tutorial \(1\)](#)
  - ⊞ [Logging and auditing Tutorial \(1\)](#)
  - ⊞ [Metrics Tutorial \(2\)](#)
  - ⊞ [Orika Mapping Tutorial \(1\)](#)
  - ⊞ [Pentaho Kettle Tutorial \(1\)](#)
  - ⊞ [Unix Tutorial \(6\)](#)

- ♥ [14 Unix interview questions & answers](#)
- ⊞ [Shell script interview questions & answers](#)
- ⊞ [SSH Login with public key](#)

“temp” folder, the present working directory (i.e. pwd) will be printed out.

**Q8** How would you go about the following scenario — you had to move to another directory temporarily to look at a file, and then move back to the directory where you were?

**A8** One way is to when you are already in /tmp folder

```
1 $ cd ../projects/JMeter
2 $ cd ../../tmp
```

The better way is to use the “**pushd**” and “**popd**” commands. These commands make use of a “stack” data structure using the “Last In First Out” approach. when you are already in /tmp folder

changes to the /projects/JMeter folder and prints the stack

```
1 $ pushd ../projects/JMeter
```

The /projects/JMeter will be popped out of the stack and the directory will change back to /tmp.

```
1 $ popd
```

If you want pushd to not print the stack, you could direct the output to the black hole /dev/null as shown below.

```
1 $ pushd ../projects/JMeter > /dev/null
```

The above is a trivial example, but in real life, you may want to navigate between more number of directories and this stack based approach will come in very handy without having to use the “cd” command. Also, very useful in Shell scripts. Use it astutely without having to build up a gigantic directory stack full of useless directories.

- Unix remoting in
- Unix Sed comm
- Unix tutorial with
- WebSpeher MQ Tu
- YAML Tutorial (1)

## 100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- Can you write code? (
- Complete the given
- Converting from A to I
- Designing your classe
- Java Data Structures
- Passing the unit tests
- What is wrong with th
- Writing Code Home A
- Written Test Core Jav
- Written Test JEE (1)

## How good are your .....?

[open all](#) | [close all](#)

- Career Making Know-
- Job Hunting & Resurr

**Q9** In Unix, only nine command line arguments can be accessed using positional parameters. How would you go about having access to more than 9 arguments?

**A9** Using the “shift” command. For example, in unix, when you run a command like

```
1 $ sh test-bash.sh file1 file2 file3 file4 file5 f
```

The \${0} is test-bash.sh, and \${1} is file1, \${2} file2 and so on till \${9}, which is file9. In the program, if you want to access file10 after processing file1 to file9, you need to use the “**shift**” command.

```
1 shift
2 l_cFile10=${9}
3 shift
4 l_cFile11=${9}
```

All it does is move all the command line arguments to the left by 1 position. Which means the file1 will be moved out, and file2 becomes \${1} and file10 becomes \${2}. If you shift it again, the file3 becomes \${1} and file11 becomes \${9}. In a nutshell

```
1 ${#} - Total number of arguments
2 ${0} - Command or the script name
3 ${1}, ${2}, ${3} - First, second and third args respectively
4 ${*} - All the command line arguments starting from the first
5 {@} - Same as ${*} except when it is quoted "${@}"
```

**Q10** How will you empty or clear the contents of a file?

**A10** Direct /dev/null (i.e. black hole) to the file

```
1 $ /dev/null > /tmp/simple-out.csv
```

**Q11** What does if [ \$? -eq 0 ] mean in a shell script?

**A11** \$? returns the status code of the last command. 0 means success and other numbers mean failure exit codes.

**Q12** How will you go about concatenating the account numbers in a number of text files to a single file?

Let's say we have a accounts.patch file with 4 files containing account numbers.

```
1 ./output/account_file_1.txt
2 ./output/account_file_1.txt
3 ./output/account_file_1.txt
4 ./output/account_file_1.txt
5
```

**A12** Firstly, cd to the folder where the “accounts.patch” file is, and then type the following command on a shell command line.

```
1 for txt-file in $(sed -n '1,4p' accounts.patch);
```

Explanation:

```
1 1, 4p
2 for sql in $(sed -n '1,4p' account_sqls.patch);
3 do cat $txt-file >> accounts_combined.txt
4
```

**Q13** How will you go about executing multiple maven commands on unix? For example, build two separate projects one after another, for example — project1 and project2. The project2 to should only build if project1 successfully builds.

**A13** Use the **&& control operator** to combine two commands so that the second is run only if the first command returns a zero exit status. In other words, if the first command runs successfully, the second command runs. If the first command fails, the second command does not run at all. For example:

```
1 $ mvn --file project1/pom.xml clean install -Dmav
```

Similarly, the **|| control operator** separates two commands and runs the second command only if the first command returns a non-zero exit status, that is if it fails. In other words,

if the first command is successful, the second command does not run. This operator is often used when testing for whether a given directory exists and, if not, it creates one.

```
1 cd project1/src/main || mkdir -p project1/src/mai
```

the -p command allows you to create any relevant parent directories as well. The && and || control characters can be used together.

Another control character that separates commands is “;”.

**Q14** How will you go about listing all the users' cron jobs?

**A14** If you are a root user, you can list the cron jobs for all the users as shown below:

```
1 for user in $(cut -f1 -d: /etc/passwd); do echo $
```

To edit cron jobs

```
1 crontab -e
```

To list the jobs

```
1 crontab -l
```

## Popular Member Posts

01: ♦ 15+ Hibernate basics Q1 – Q7 interview questions & answers

1,754 views

♦ 11 Spring boot interview questions & answers

912 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

823 views

01: ♦ 15 Beginner level Java multi-threading interview Q&A

521 views

001A: ♦ 7+ Java integration styles & patterns  
interview questions & answers

421 views

18 Java scenarios based interview Questions and  
Answers

389 views

♦ 7 Java debugging interview questions & answers

318 views

01: ♦ 15 Ice breaker questions asked 90% of the time  
in Java job interviews with hints

297 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview  
Questions and Answers

294 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions  
& answers

294 views

Bio

Latest Posts



## Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. Published Java interview Q&A books via [Amazon.com](http://Amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#) to get notified of special offers. User & expert [reviews](#).





**About** [Arulkumaran Kumaraswamipillai](#)

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. Published Java interview Q&A books via [Amazon.com](#) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#) to get notified of special offers. User & expert [reviews](#).

◀ 01: Java data types interview Q&A

Shell script interview Q&A for Java developers ▶

**Posted in** Tools, Unix Interview Q&A, Unix Tutorial

**Tags:** Free Content, Java/JEE FAQs

## Leave a Reply

Your email address will not be published. Required fields are marked \*

### Comment

### Name \*

### E-mail \*

**Website**

## Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work?](#)

[What is a hashing function?](#)

☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

### Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)

## Prepare to succeed

☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

## © Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.