

Multidimensional Array in awk

Spoken Tutorial Project

<http://spoken-tutorial.org>

National Mission on Education through ICT

<http://sakshat.ac.in>

Antara Roy Choudhury

Praveen S

IIT Bombay

10 January 2018



Learning Objectives

We will learn to



Learning Objectives

We will learn to

- Create multidimensional array in awk



Learning Objectives

We will learn to

- Create multidimensional array in awk
- Scan the multidimensional array



System Requirements



System Requirements

- **Ubuntu Linux 16.04 OS**



System Requirements

- **Ubuntu Linux 16.04 OS**
- **gedit text editor 3.20.1**



Pre-requisites

- Please go through previous awk tutorials on array on <http://spoken-tutorial.org/>



Pre-requisites

- Please go through previous awk tutorials on array on <http://spoken-tutorial.org/>
- Knowledge of any programming language like C or C++



Code Files

- The files used in this tutorial are available in the Code Files link on this tutorial page
- Please download and extract them



Single dimensional Array in awk



Single dimensional Array in awk

- **Single dimensional array element is identified by a single index**



Single dimensional Array in awk

- Single dimensional array element is identified by a single index
- For example `week[day]`



Multidimensional Array in awk

Multidimensional array-

- An element is identified by
sequence of multiple indices



Multidimensional Array in awk

Multidimensional array-

- An element is identified by **sequence of multiple indices**
- A two dimensional array element is identified by a sequence of 2 indices



Multidimensional Array(Cont.)

- Multiple indices are concatenated into one string



Multidimensional Array(Cont.)

- Multiple indices are concatenated into one string
- A separator between them



Multidimensional Array(Cont.)

- Multiple indices are concatenated into one string
- A separator between them
- **Separator is the value of the built-in variable SUBSEP**



Multidimensional Array(Cont.)

- This combined string is used as a single index for a simple one dimensional array



Example

- `multi[4,6] = "value"`



Example

- `multi[4,6] = "value"`
- Numbers 4 and 6 are converted to string



Example

- `multi[4,6] = "value"`
- Numbers 4 and 6 are converted to string
- **SUBSEP** is `#`



Example

- `multi[4,6] = "value"`
- Numbers 4 and 6 are converted to string
- SUBSEP is `#`
- Those numbers are concatenated with `#` between them



Example(Cont.)

- `multi["4#6"]` is set to "value"



Example(Cont.)

- `multi["4#6"]` is set to "value"
- Default value of **SUBSEP** is the string `"\034"`



Example(Cont.)

- `multi["4#6"]` is set to "value"
- Default value of SUBSEP is the string `"\034"`
- Is actually a non-printing character



Example(Cont.)

- `multi["4#6"]` is set to "value"
- Default value of SUBSEP is the string `"\034"`
- Is actually a non-printing character
- It will not appear usually in most input data



Example(Cont.)

	Col1	Col2
Row1	A	B
Row2	C	D



Test of presence

- Test if a particular index sequence exists in a given multidimensional array?



Test of presence

- Test if a particular index sequence exists in a given multidimensional array?
- Use **in** operator



Test of presence

- Test if a particular index sequence exists in a given multidimensional array?
- Use **in operator**
- Write entire sequence of indices within parentheses



Test of presence

- Test if a particular index sequence exists in a given multidimensional array?
- Use **in operator**
- Write entire sequence of indices within parentheses
- **Separated by commas**



Transpose of a matrix

- We want to create the transpose of a matrix



Transpose of a matrix

- We want to create the transpose of a matrix
- It is formed by interchanging the rows and columns of a matrix



Explanation

- **Process first record to last record**



Explanation

- Process first record to last record
- For 1st record, `max_nr=1`



Explanation

- Process first record to last record
- For 1st record, `max_nr=1`
- For 2nd record, `max_nr=2`



Explanation

- Process first record to last record
- For 1st record, `max_nr=1`
- For 2nd record, `max_nr=2`
- For last record, `max_nr=total number of records`



Scanning multidimensional array

- **Awk does not have multidimensional array**



Scanning multidimensional array

- Awk does not have multidimensional array
- No special **for statement** to scan the multidimensional array



Scanning multidimensional array

- Awk does not have multidimensional array
- No special **for statement** to scan the multidimensional array
- **Multidimensional way of scanning array**



Scanning multidimensional array

- Awk does not have multidimensional array
- No special **for statement** to scan the multidimensional array
- Multidimensional way of scanning array
- **Combine for statement with split()**



Split function



Split function

- **split function is used to chop up or split a string into pieces**



Split function

- split function is used to chop up or split a string into pieces
- Place the various pieces into an array



Split function Syntax

```
split(string, arr, sep)
```



Split function Syntax

```
split(string, arr, sep)
```

- **First argument contains the string to be chopped**



Split function Syntax

```
split(string, arr, sep)
```

- **First argument contains the string to be chopped**
- **Second argument specifies the name of the array where split will put the chopped pieces into**



Split function Syntax(Cont.)

```
split(string, arr, sep)
```

- **The third argument mentions the separator that will be used to chop the string up**



Split function Syntax(Cont.)

```
split(string, arr, sep)
```

- The third argument mentions the separator that will be used to chop the string up
- **arr[1]=first piece**



Split function Syntax(Cont.)

```
split(string, arr, sep)
```

- The third argument mentions the separator that will be used to chop the string up
- `arr[1]`=first piece
- `arr[2]`=second piece, so on



Summary

We learnt to-

- Create a multidimensional array in awk
- Scan a multidimensional array



Assignment

Write an awk script to

- **rotate a two dimensional array by 90 degree**
- **print the rotated matrix**



About the Spoken Tutorial Project

- Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- It summarizes the Spoken Tutorial project
- If you do not have good bandwidth, you can download and watch it



Spoken Tutorial Workshops

The Spoken Tutorial Project Team

- Conducts workshops using spoken tutorials
- Gives certificates to those who pass an online test
- For more details, please write to contact@spoken-tutorial.org



Forum for specific questions

- Do you have questions in **THIS Spoken Tutorial?**
- Please visit
<http://forums.spoken-tutorial.org>
- Choose the minute and second where you have the question
- Explain your question briefly
- Someone from our team will answer them



Acknowledgements

- Spoken Tutorial Project is a part of the Talk to a Teacher project
- It is supported by the National Mission on Education through ICT, MHRD, Government of India
- More information on this Mission is available at:

<http://spoken-tutorial.org/NMEICT-Intro>



Thank You

Thanks for joining
<http://spoken-tutorial.org>

