OPERATING SYSTEMS

UE22CS242B

awk

Suresh Jamadagni
Department of Computer Science

OPERATING SYSTEMS

awk

Suresh Jamadagni
Department of Computer Science

Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau
5. Internet source

AWK

- Awk is a scripting language used for manipulating data and generating reports.
- The awk command programming language requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators.
- Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line.
- Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then perform the associated actions.

AWK

Operations:

- Scan a file line by line
- Split each input line into fields
- Compare input line/fields to pattern
- Perform action(s) on matched lines

Programming Constructs Provided:

- Format output lines
- Arithmetic and string operations
- Conditionals and loops

AWK syntax

awk options 'selection _criteria {action }' input-file > output-file

Options:

- -f program-file : Reads the AWK program source from the file program-file, instead of from the first command line argument.
- -F fs : Use fs for the input field separator

AWK examples

Print selected columns from a file

```
awk '{print $1,$4}' employee.txt
```

Print lines matching a criteria

```
awk '/manager/ {print}' employee.txt
awk '{if($4 >= "30000") print $1,$4;}' employee.txt
awk '{if($4 >= "30000" && $2 == "manager") print $1,$2,$4;}' employee.txt
awk '{if($4 >= "30000" && $2 != "manager") print $1,$2,$4;}' employee.txt
awk '{if($4 >= "30000" || $2 == "manager") print $1,$2,$4;}' employee.txt
```

AWK examples

Print total number of lines in a file

```
awk 'END { print NR }' employee.txt
```

Print total number of lines matching a criteria

```
awk '{if($4 >= "30000") count++} END {print count}' employee.txt
```

Using built-in functions to change the output

```
awk '/manager/ {print substr($1,1,4)}' employee.txt
awk '/manager/ {print toupper($1)}' employee.txt
awk '/manager/ {print tolower($1)}' employee.txt
awk '/manager/ {print toupper(substr($1,1,1)) substr($1,2)}' employee.txt
awk '{if($4 >= "30000" || $2 == "manager") print $1,$2,$4,$4*1.1;}' employee.txt
```

PES UNIVERSITY
CELEBRATING 50 YEARS

THANK YOU

Suresh Jamadagni
Department of Computer Science Engineering

sureshjamadagni@pes.edu