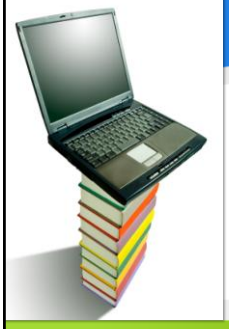


Jerzy Nawrocki
Faculty of Computing & Information Sci.
Poznan University of Technology
jerzy.nawrocki@put.poznan.pl

Text processing and AWK

File conversion problem

```
FName:John   SName:Great   Salary 585
FName:Ann    SName:Nice    Salary 700
```

FName	SName	Salary
John	Great	585
Ann	Nice	700

File conversion problem

```
#include <stdio.h>
#include <stdlib.h>
FILE *fin;
char token[200];
char gettoken(void)
{int i=0;
char c;
do {c = getc(fin);
    if (c == EOF) return (EOF);
    } while (c < '!');
```

Solution in C:

≈ 40 lines of code

File conversion problem

Solution in AWK

```
BEGIN   {FS=":| ";}
NR == 1 {print $1, "\t", $3, "\t", $5;}
        {gsub(/,/, "." , $6); print $2, "\t", $4, "\t", $6;}
```


Origins of AWK

Bell Labs, Murray Hill (New Jersey), Foto: http://en.wikipedia.org/wiki/Bell_Labs

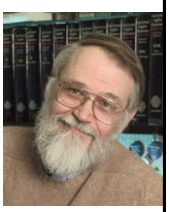
Bell Labs, New Jersey (USA), 1977

AWK: Aho, Weinberger, Kernighan
Platforms: Unix, MS DOS/Windows
Similarity to C

Authors of AWK

Alfred Aho

Peter Weinberger



Brian Kernighan

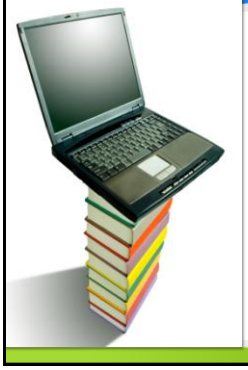
http://www.underforty.us/geeks.html

Aim of the lecture

To present:

- **Another programming paradigm (rule-based programming)**
- **Regular expressions**
- **Basics of AWK**

Agenda



- **Fundamentals of AWK**
- **Simplest programs**
- **Patterns**
- **Regular expressions**
- **Variables**

Fundamental question

Input file

Field

Line

```
Jerzy Nawrocki    43089  I1
Jane  Kowalski    43780  I2
Adam  Malinowski  43990  I1
```

Fields: **$1**, **$2**, **$3**, ...

Program structure

Processing rule

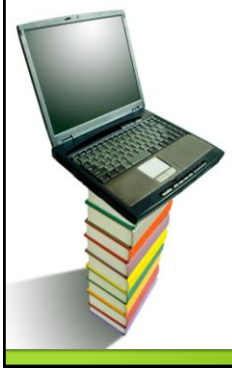
```
pattern1   {instruction1}
pattern2   {instruction2}
  ...          ...
```


Execution principle

```
Jerzy Nawrocki
Jane  Kowalski
Adam  Malinowski
```

```
pattern1 {instruction1}
pattern2 {instruction2}
  ...         ...
```

Agenda



- Fundamentals of AWK
- Simplest programs
- Patterns
- Regular expressions
- Variables

Simplest programs

How many fields on the output?

```
Jerzy Nawrocki    43089  I1
Jane  Kowalski    43780  I2
Adam  Malinowski  43990  I1
```

```
$4=="I1"      { print $2, $1; }
```

```
Nawrocki Jerzy
Malinowski Adam
```


Simplest programs

How many fields on the output?

```
Jerzy Nawrocki    43089  I1
Jane  Kowalski    43780  I2
Adam  Malinowski  43990  I1
```

```
$4=="I1"
```

```
Jerzy Nawrocki    43089  I1
Adam  Malinowski  43990  I1
```


Simplest programs

What field will be first?

```
Jerzy Nawrocki    43089  I1
Jane  Kowalski    43780  I2
Adam  Malinowski  43990  I1
```

```
{ print $2, $1; }
```

```
Nawrocki Jerzy
Kowalski Jane
Malinowski Adam
```


Agenda

- Fundamentals of AWK
- Simplest programs
- Patterns
- Regular expressions
- Variables

Patterns

- Begining and end of text
- Relations
- Compound patterns
- ~~Range patterns~~
- *Regular expressions*

Begining and end of text

```
Jerzy Nawrocki    43089   I1
Jane  Kowalski    43780   I2
Adam  Malinowski  43990   I1
```

```
BEGIN       { print "-----"; }
$4=="I2"    { print $2, $1; }
END         { print "*****"; }
```

```
-----
Kowalski Jane
*****
```


Begining and end of text

```
Jerzy Nawrocki    43089   I1
Jane  Kowalski    43780   I2
Adam  Malinowski  43990   I1
```

```
END         { print "*****"; }
$4=="I2"    { print $2, $1; }
BEGIN       { print "-----"; }
```

```
-----
Kowalski Jane
*****
```


Relations

```
12      11
 2      11
```

```
$1 > $2
```

```
12      11
```


Compound patterns

`||`	or
	`$1==1 || $2==1`
`&&`	and
	`$1==1 && $2==1`
`!`	not
	`! $1==1`

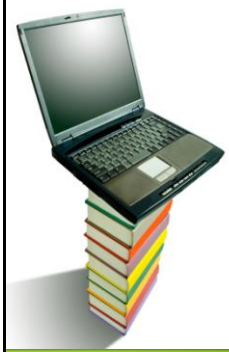
Compound patterns

```
Jerzy Adam        43089   I1
Adam  Kowalski    43780   I2
Adam  Malinowski  43990   I1
```

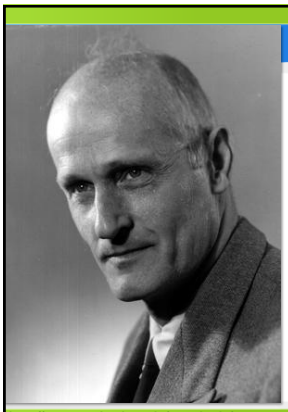
```
$4=="I1" && $1=="Adam"    { print $2, $1; }
```

```
Malinowski Adam
```


Agenda



- Fundamentals of AWK
- Simplest programs
- Patterns
- **Regular expressions**
- Variables



Stephen Kleene

1909-01-05, Connecticut, USA

1934: Dr, Princeton Univ., (Alonzo Church)

1935: Univ. of Wisconsin-Madison (USA)

1939-40: Inst. for Advanced Study, Princeton – recursion theory

1990: National Medal of Sci.

1994-01-25, Madison

http://www.math.wisc.edu/~gpslogic/

Regular expressions

Arithmetic expressions

Value: Text → Number

Value(2·3 + 3) = 9

Regular expressions

Value: Text → SetOfCharacterStrings

Value(/Ala | Ola/) = {"Ala", "Ola"}

Patterns with regular expressions

e.g. a character or string

http://www.freewebs.com/limericks/ by Terry Walsh

$0, $1, $2, ..

Whole string `String ~ /^ reg_exp $/`

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

$1 ~ /^I$/

Patterns with regular expressions

e.g. a character or string

http://www.freewebs.com/limericks/ by Terry Walsh

$0, $1, $2, ..

Whole string `String ~ /^ reg_exp $/`

Begining `String ~ /^ reg_exp /`

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

$1 ~ /^I/

Patterns with regular expressions

e.g. a character or string

http://www.freewebs.com/limericks/ by Terry Walsh

$0, $1, $2, ..

Whole string `String ~ /^ reg_exp $/`

Begining `String ~ /^ reg_exp /`

End `String ~ / reg_exp $/`

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

$3 ~ /e$/

Patterns with regular expressions

e.g. a character or string

http://www.freewebs.com/limericks/ by Terry Walsh

$0, $1, $2, ..

Whole string `String ~ /^ reg_exp $/`

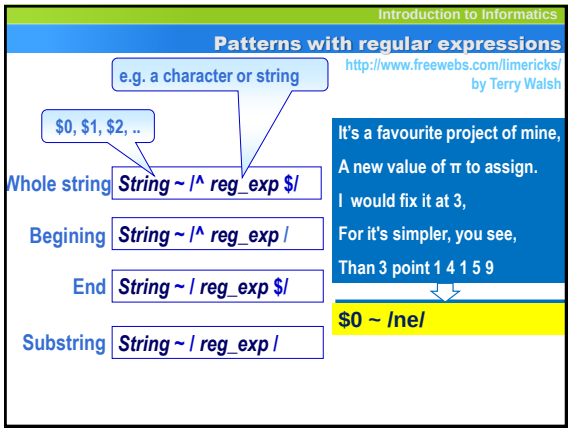
Begining `String ~ /^ reg_exp /`

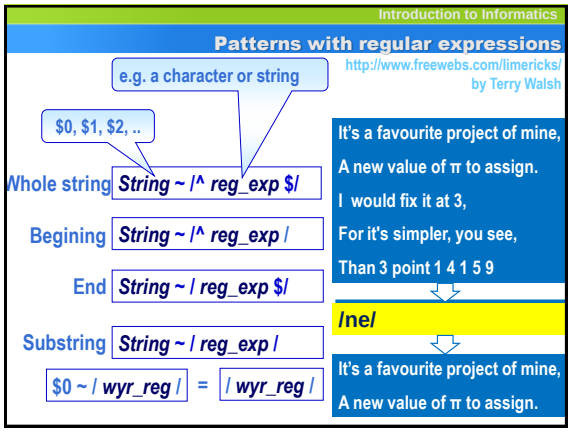
End `String ~ / reg_exp $/`

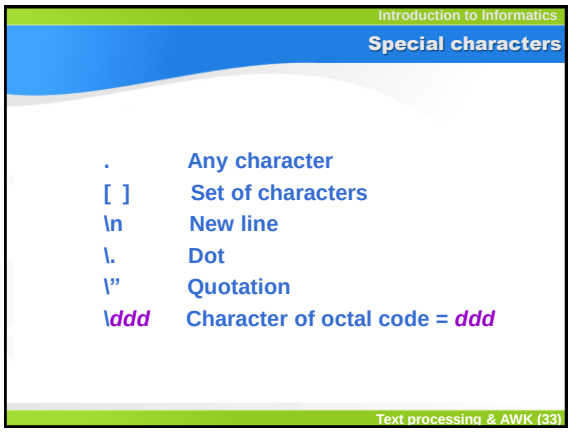
Substring `String ~ / reg_exp /`

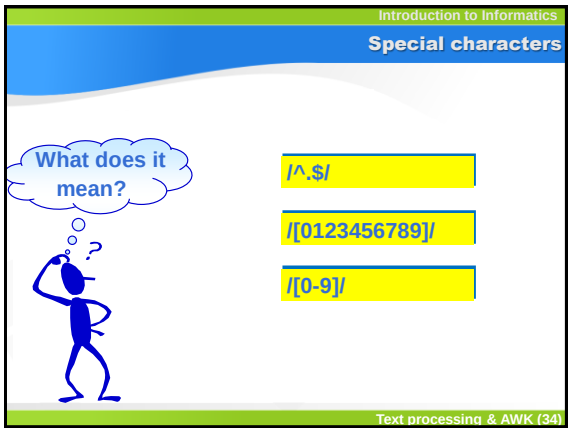
It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

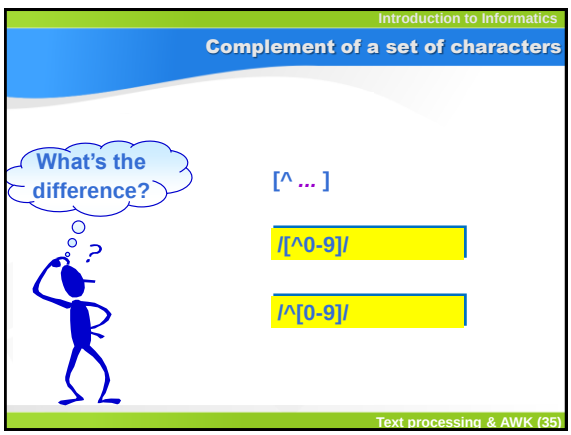
$3 ~ /e/

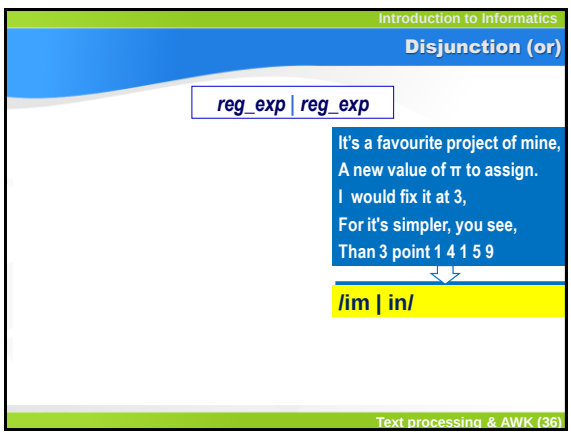

Introduction to Informatics
Patterns with regular expressions
e.g. a character or string
http://www.freewebs.com/limericks/
by Terry Walsh
$0, $1, $2, ..
Whole string String ~ /^ reg_exp $/
Begining String ~ /^ reg_exp /
End String ~ / reg_exp $/
Substring String ~ / reg_exp /
It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9
$0 ~ /ne/


Introduction to Informatics
Patterns with regular expressions
e.g. a character or string
http://www.freewebs.com/limericks/
by Terry Walsh
$0, $1, $2, ..
Whole string String ~ /^ reg_exp $/
Begining String ~ /^ reg_exp /
End String ~ / reg_exp $/
Substring String ~ / reg_exp /
$0 ~ / wyr_reg / = / wyr_reg /
It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9
/ne/
It's a favourite project of mine,
A new value of π to assign.


Introduction to Informatics
Special characters
. Any character
[ ] Set of characters
\n New line
\. Dot
\" Quotation
\ddd Character of octal code = ddd
Text processing & AWK (33)


Introduction to Informatics
Special characters
What does it mean?
/^.$/
/[0123456789]/
/[0-9]/
Text processing & AWK (34)


Introduction to Informatics
Complement of a set of characters
What's the difference?
[^ ...]
/[^0-9]/
/^[0-9]/
Text processing & AWK (35)


Introduction to Informatics
Disjunction (or)
reg_exp | reg_exp
It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9
/im | in/
Text processing & AWK (36)

Parentheses and disjunction

Parentheses change precedence of operators:
3*(4 + 5) = 3* 9 = 27

Distributive law:
3*(4 + 5) = 3*4 + 3*5 = 12 + 15 = 27
(4 + 5)*3 = 4*3 + 5*3 = 12 +15 = 27

Distributivity of concatenation over disjunction:

/i(m | n)/ = /im | in/

/Lond(o | y)n/ = /London | Londyn/

Kleene's star

Select all lines for which the first field is a number.

```
1 2 3
AD 1984
4 Ladies
```

```
$1 ~ /[0-9]/
```


Kleene's star

Select all lines for which the first field is a number.

```
1 2 3
AD 1984
4 Ladies
```

```
4tune:
2 cents
```

```
$1 ~ /[0-9]/
```

```
1 2 3
4 Ladies
```

```
4tune:
2 cents
```


Kleene's star

Select all lines for which the first field is a 1-digit number.

```
1 2 3
AD 1984
4 Ladies
```

```
4tune:
2 cents
```

```
4tune:
32 cents
0
```

```
$1 ~ /^ [0-9] $/
```


Kleene's star

Select all lines for which the first field is a 1- or 2-digit number.

```
1 2 3
AD1984
4 Ladies
```

```
4tune:
2 cents
```

```
4tune:
32 cents
0
```

```
$1 ~ /^( [0-9] | [0-9][0-9] )$/
```

1 digit 2 digits

Kleene's star

Select all lines for which the first field is a 1- or 2-digit number.

```
1 2 3
AD 1984
4 Ladies
```

```
4tune:
2 cents
```

```
4tune:
32 cents
0
```

```
4tune:
32 cents
120 euro
```

```
$1 ~ /^( [0-9] | [0-9][0-9] )$/
```

```
1 2 3
4 Ladies
```

```
2 cents
```

```
32 cents
0
```

Kleene's star

Select all lines for which the first field is a number.

1 2 3 AD 1984 4 Ladies	4tune: 2 cents	4tune: 32 cents 0	4tune: 32 cents 120 euro

```
$1 ~ /^( [0-9] | [0-9][0-9] | [0-9][0-9][0-9] )$/
```

1 digit 2 digits 3 digits

Kleene's star

$[0\text{-}9] = [0\text{-}9]^1$

$[0\text{-}9][0\text{-}9] = [0\text{-}9]^2$

$[0\text{-}9][0\text{-}9][0\text{-}9] = [0\text{-}9]^3$

. . . . . .

$[0\text{-}9]^1 \mid [0\text{-}9]^2 \mid [0\text{-}9]^3 \mid \ldots = [0\text{-}9]+$

Kleene's star

Select all lines for which the first field is a number.

1 2 3 AD 1984 4 Ladies	4tune: 2 cents	4tune: 32 cents 0	4tune: 32 cents 120 euro

```
$1 ~ /^ [0-9]+ $/
```

1 2 3 4 Ladies	2 cents	32 cents 0	32 cents 120 euro

Kleene's star

Can be a sequence of w's.

A non-empty sequence of w's.

$w+ = w \mid ww \mid www \mid \ldots$

$w^* = \varepsilon \mid w \mid ww \mid www \mid \ldots$

Empty string

$w(\varepsilon \mid w \mid ww \mid www \mid ..) =$
$w \mid ww \mid www \mid wwww \mid ..$

$x\,\varepsilon = x$

$\varepsilon\,x = x$

$w+ = w\,w^* = w^*\,w$

$x\,w^* = x \mid x\,w+$

$w^*x = x \mid w+x$

Riddle

What's its meaning?

```
$2 ~ /^[0-9][0-9]*$/
```

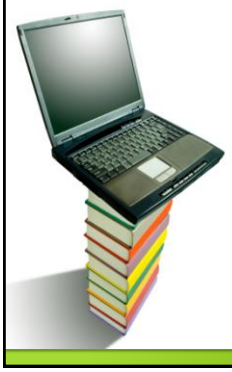

Riddle

What on the output?

```
Payment  2010/11
================
Nawrocki 160
Antczak  359
```

```
$2 ~ /^[0-9]+$/
```

Agenda



- Fundamentals of AWK
- Simplest programs
- Patterns
- Regular expressions
- Variables

Variables

- Variables introduced by a programmer
  (*type*: string of characters;
  initial value: empty string / zero)
- Built-in variables
  (standard meaning)
- Field variables $1, $(i+j-1), ..

Some built-in variables

NF – number of fields in a row
NR – row number
FILENAME – file name with input data

Variables

NR	NF	total
1	2	0
2	1	2
		3

```
If you
have
a hammer, everything
looks like a nail
```

```
{total= total + NF;}
END {print "Fields: ", total;
     print "Rows: ", NR;}
```


Summary

Input file

Field

Line

```
Jerzy Nawrocki    43089   I1
Jane  Kowalski    43780   I2
Adam  Malinowski  43990   I1
```

Fields: $1, $2, $3, ...

Program structure

Processing rule

```
pattern1   {instruction1}
pattern2   {instruction2}
  ...          ...
```


Execution principle

```
Jerzy Nawrocki
Jane  Kowalski
Adam  Malinowski
```

```
pattern1 {instruction1}
pattern2 {instruction2}
  ...          ...
```


Regular expressions

$\varepsilon\, x = x\, \varepsilon$

$L_1 \bullet L_2 = \{xy: x \in L_1 \wedge y \in L_2\}$

$L^0(r) = \{\varepsilon\}$

$L^{n+1}(r) = L(r) \bullet L^n(r)$

$L^n(r) = \{\underbrace{xx\ldots x}_{n}: \; x \in L(r)\}$

$L(r^*) = \bigcup_{n=0}^{\infty} L^n(r)$

$(01)^* = \{\varepsilon, 01, 0101, 010101, \ldots\}$

$L(r^*) = \{\varepsilon, x, xx, xxx, \ldots: \; x \in L(r)\}$

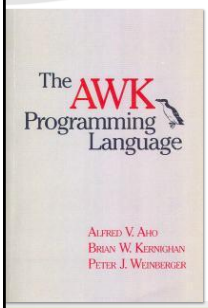
Summary

gawk -f prog.awk <in.txt >out.txt

Other features of AWK:

- Compound instructions (if, while, ..)
- Dynamic arrays
- Built-in functions (gsub, ..)

Literature



- A. Aho, B. Kernighan, P. Weinberger, *The AWK Programming Language*, Addison-Wesley, Reading, 1988.
- J. Nawrocki, W. Complak, Wprowadzenie do przetwarzania tekstów w języku AWK, *Pro Dialog* 2 (1994), 23-46.
- J. Cybulka, B. Jankowska, J.R. Nawrocki, *Automatyczne przetwarzanie tekstów. AWK, Lex i YACC*, Nakom, Poznań, 2002.