awk

A language for processing fields and records

Introduction

- It is a programming language
- awk is an abbreviation of the three people who developed it: Aho, Weinberger & Kernighan
- It is a part of POSIX, IEEE 1003.1-2008
- Variants: nawk, gawk, mawk ...
- gawk contains features that extend POSIX

Execution model

- Input stream is a set of records
- Eg., using “\n” as record separator, lines are records
- Each record is a sequence of fields
- Eg., using “ “ as field separator, words are fields
- Splitting of records to fields is done automatically
- Each code block executes on one record at a time, as matched by the pattern of that block

usage

- Single line at the command line

```
cat /etc/passwd | awk -F”:” ‘{print $1}’
```

- Script interpreted by awk

```
./myscript.awk /etc/passwd
```

myscript.awk

```
#!/usr/bin/gawk -f
BEGIN {
    FS=”:”
}
{
    print $1
}
```

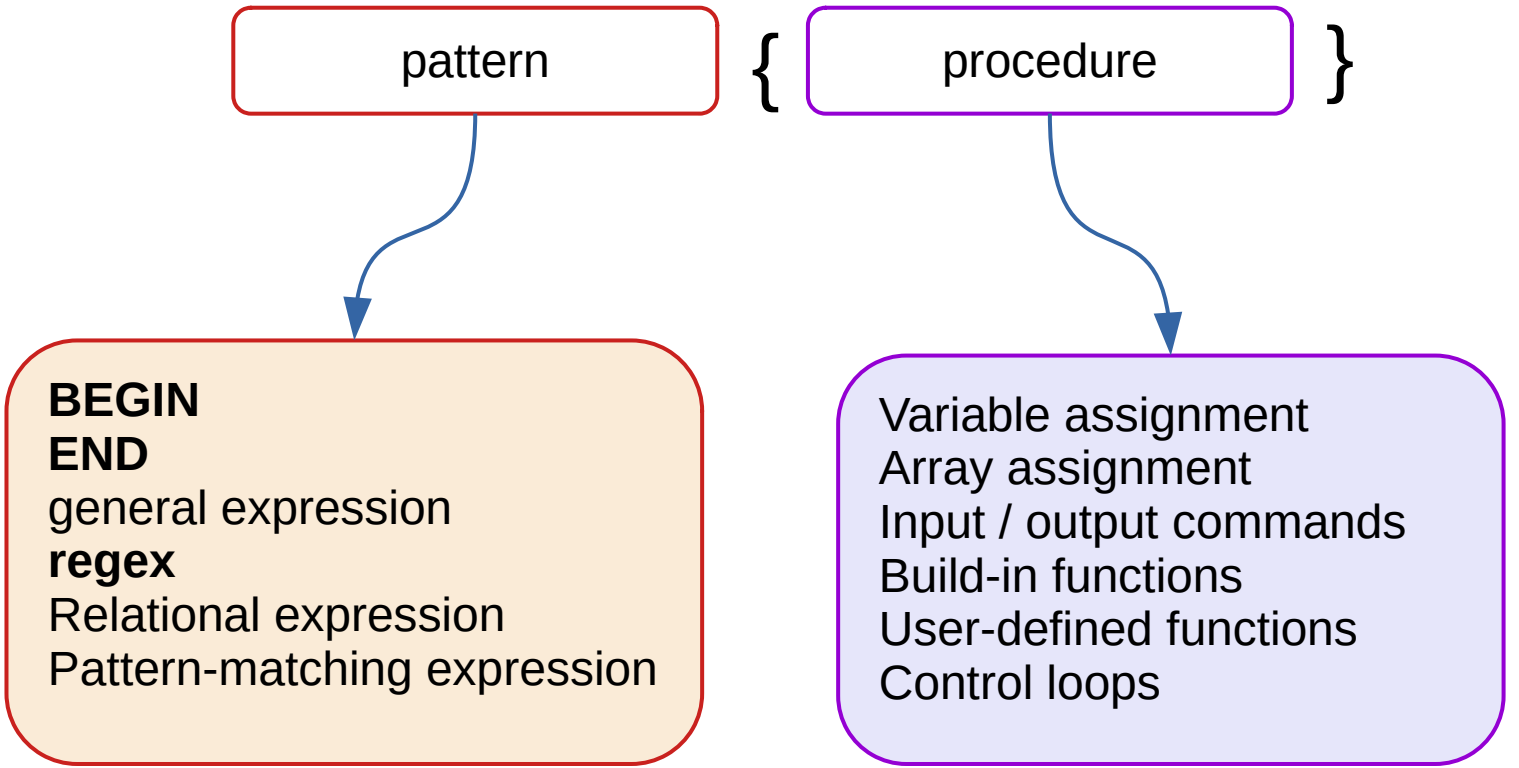
Built-in variables

ARGC	Number of arguments supplied on the command line (except those that came with -f & -v options)
ARGV	Array of command line arguments supplied; indexed from 0 to ARGC-1
ENVIRON	Associative array of environment variables
FILENAME	Current filename being processed
FNR	Number of the current record, relative to the current file
FS	Field separator, can use regex
NF	Number of fields in the current record
NR	Number of the current record
OFMT	Output format for numbers

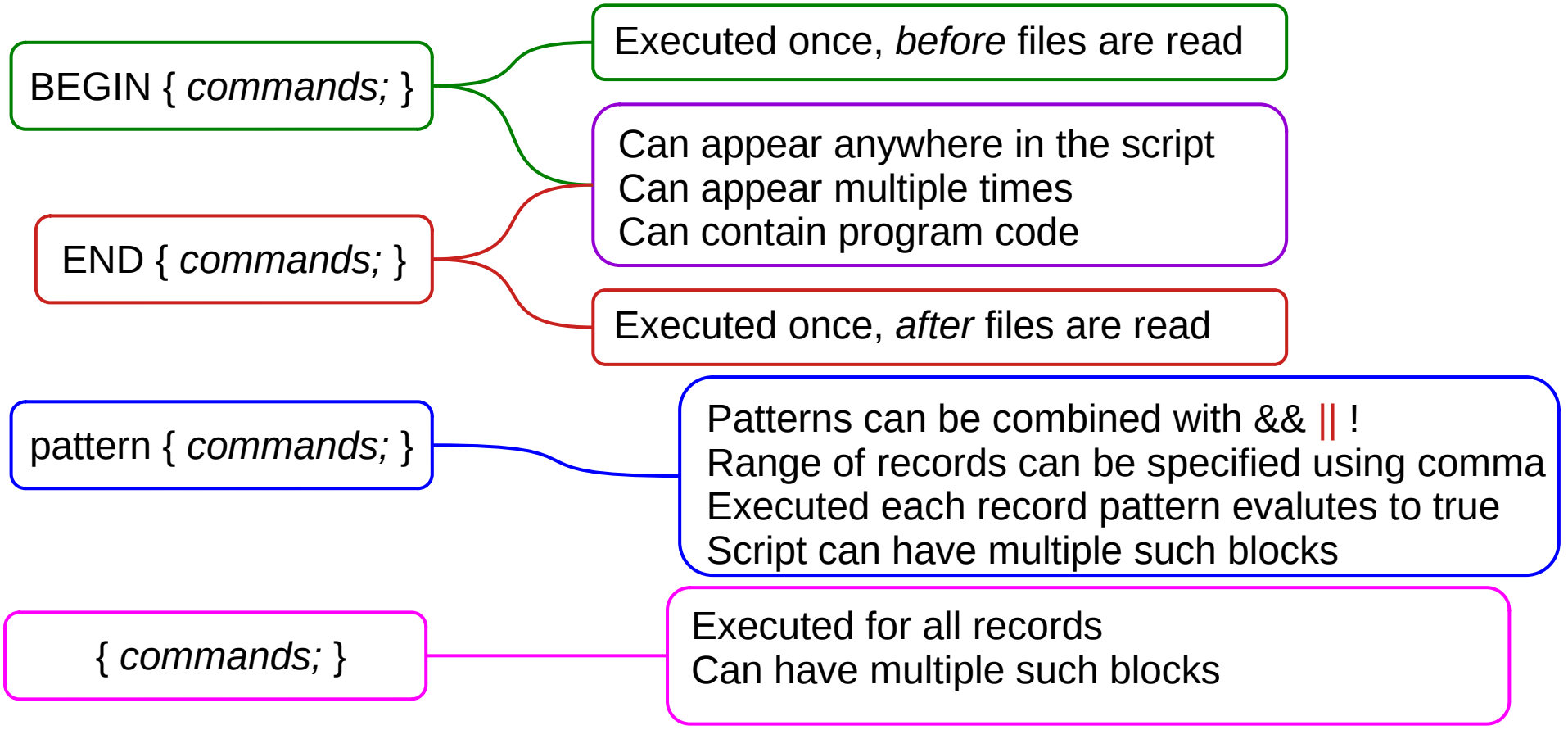
Built-in variables

OFS	Output fields separator
ORS	Output record separator
RS	Record separator
RLENGTH	Length of string matched by match() function
RSTART	First position in the string matched by match() function
SUBSEP	Separator character for array subscripts
$0	Entire input record
$n	nth field in the current record

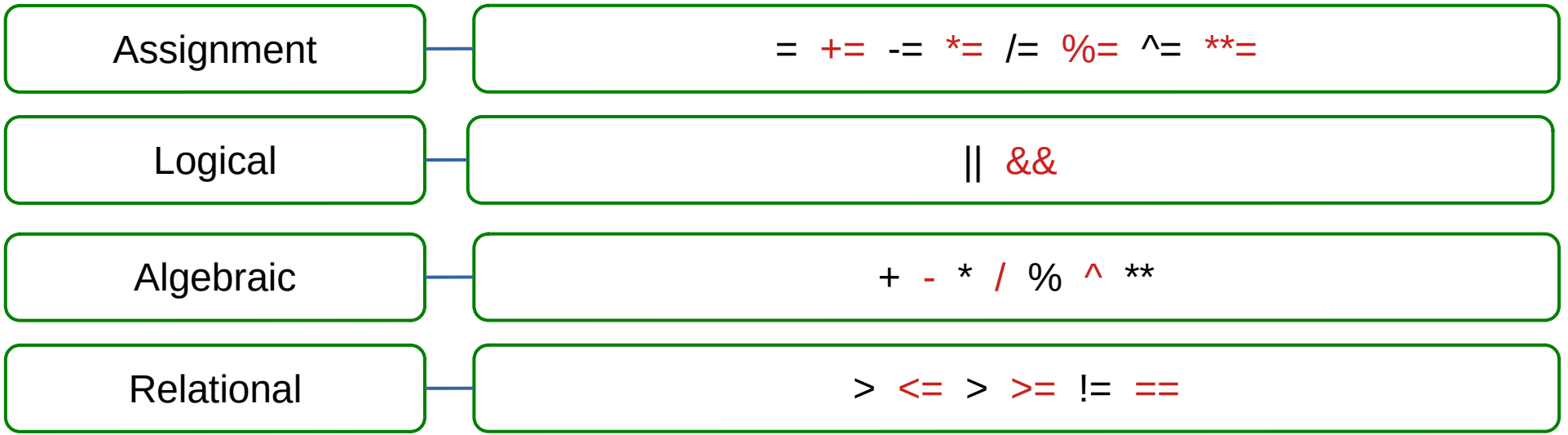
awk scripts


pattern
{
procedure
}
BEGIN
END
general expression
regex
Relational expression
Pattern-matching expression
Variable assignment
Array assignment
Input / output commands
Build-in functions
User-defined functions
Control loops

execution


BEGIN { commands; }
Executed once, before files are read
Can appear anywhere in the script
Can appear multiple times
Can contain program code
END { commands; }
Executed once, after files are read
pattern { commands; }
Patterns can be combined with && || !
Range of records can be specified using comma
Executed each record pattern evalutes to true
Script can have multiple such blocks
{ commands; }
Executed for all records
Can have multiple such blocks

operators



Operators

`expr ? a : b`	Conditional expression
`a in array`	Array membership
`a ~ /regex/`	Regular expression match
`a !~ /regex/`	Negation of regular expression match
`++`	Increment, both prefix and postfix
`--`	decrement, both prefix and postfix
`$`	Field reference
	Blank is for concatenation

Functions and commands

Arithmetic	atan2 cos exp int log rand sin sqrt srand
String	asort asorti gsub index length match split sprintf strtonum sub substr tolower toupper
Control Flow	break continue do while exit for if else return
Input / Output	close fflush getline next nextline print printf
Programming	extension delete function system
bit-wise	and compl lshift or rshift xor

arrays

- Associative arrays
- Sparse storage
- Index need not be integer
- `arr[index]=value`
- `for (var in arr)`
- `delete arr[index]`

Loops

```
for (a in array)
{
    print a
}
```

```
if (a > b)
{
    print a
}
```

```
for (i=1;i<n;i++)
{
    print i
}
```

```
while (a < n)
{
    print a
}
```

```
do
{
    print a
} while (a < n)
```

functions

```
cat infile |awk -f mylib -f myscript.awk
```

mylib

```
function myfunc1()
{
    printf “%s\n”, $1
}

function myfunc2(a)
{
    return a*rand()
}
```

myscript.awk

```
BEGIN
{
    a=1
}
{
    myfunc1()
    b = myfunc2(a)
    print b
}
```

Pretty printing

```
printf “format”, a, b, c
```

%[modifier]control-letter

width
prec
-

c	ascii char
d	integer
i	integer
e	scientific notation
f	floating notation

g	shorter of scientific & float
o	octal value
s	string text
x	hexadecimal value
X	hexadecimal value in caps

bash + awk

- Including awk inside shell script
- heredoc feature
- Use with other shell scripts on command line using pipe

awk is available everywhere !

awk is a programming language, quick to code and fast in execution

combine it on the command line with other scripts