Operációs rendszerek gyak.

AWK programozás

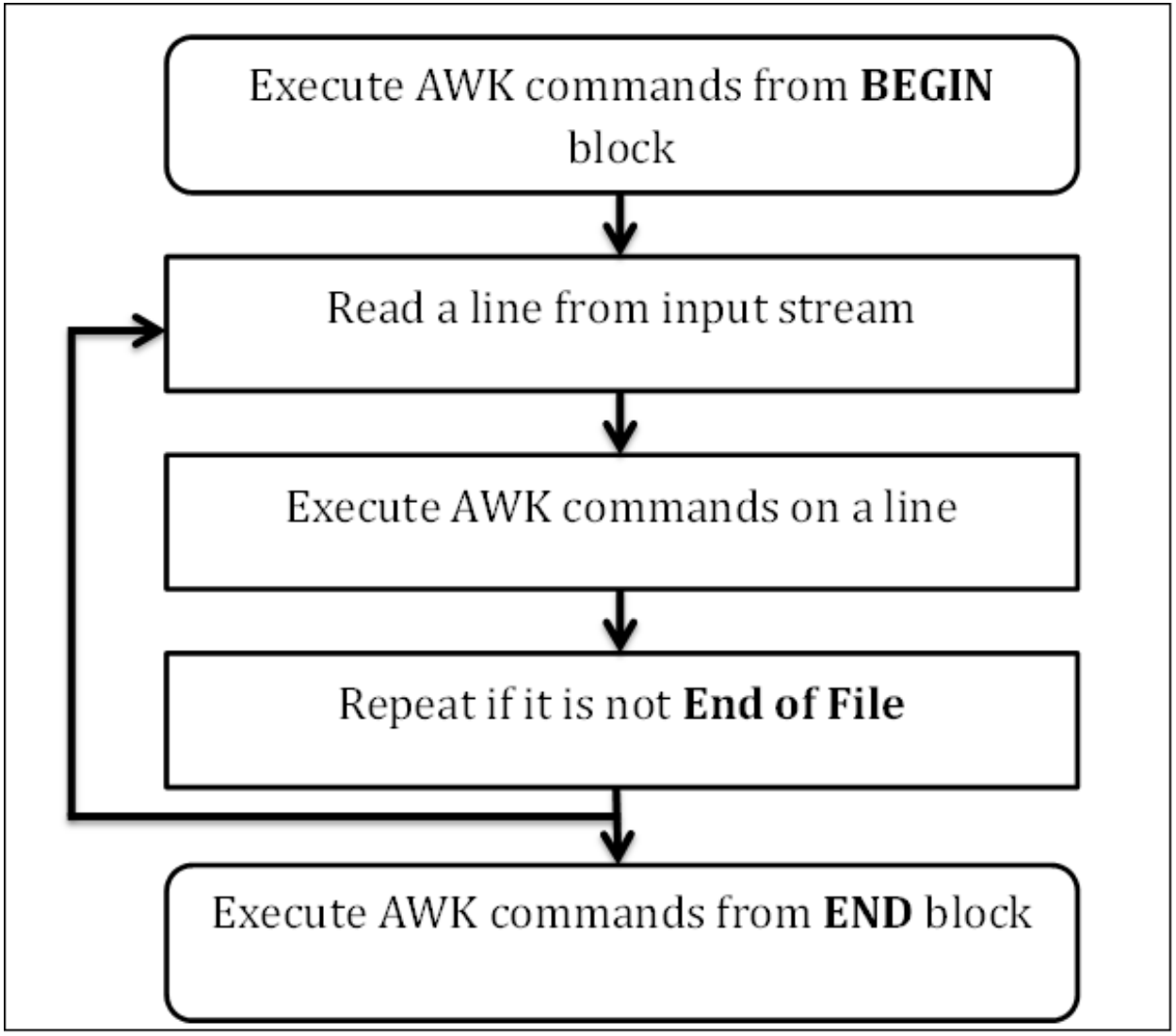
Hirling Dominik

Szegedi Tudományegyetem

AWK

- AWK: a pattern scanning and processing language
- mintaelemző-és feldolgozó nyelv
- bármilyen szövegből minták alapján sorokat, mezőket képes kiemelni és ezeken a mezőkön módosításokat végezni
- C-re épül a szerkezete

AWK felépítés



AWK futtatása

- közvetlenül a parancssorból: `awk 'PROGRAM' INPUTFILE` (nem ajánlott összetettebb feladatoknál)
- awk program forrásfájlból olvasva: `awk -f PROGRAMFAJL INPUTFILE`
- futtatható awk scriptként, ekkor:
  - parancsértelmező fejlécbe `#!/usr/bin/awk -f`
  - futtatás ./scriptfile INPUTFILE

AWK program felépítése

- minden awk program **szabályok** sorozata
- minden szabály tartalmazhat egy **mintát** és egy hozzá tartozó **akciót**
- minden akció **utasítások**ból áll
- a szabályok alakja: `MINTA {AKCIÓ}`
- a szabályokat (és az utasításokat is) egymástól sortöréssel vagy pontosvesszővel választhatjuk el

AWK program váz

BEGIN{

. . .

}

SZABALY1{

. . .

}

SZABALY2{

. . . .

}

. . .

END{

. . .

}

AWK program futása

- feldolgozás során a program a bemenet tartalmát **rekordok**ra (records) bontja, amik alapesetben a bemenet sorai
- a rekordokat a program tovább bontja **mezők**re (fields), amik alapesetben a rekordok szóközzel elválasztott szavai
- a bemenet rekordonként kerül feldolgozásra, a program megpróbálja az aktuális rekordot illeszteni az összes szabály mintájára
- ha a bemenet illeszkedik egy szabály mintájára, akkor végrehajtódik a hozzá tartozó akció
- a mintákban nem csak a rekord tartalmától függő feltételeket fogalmazhatunk meg
- ha egy szabályhoz nincs minta hozzárendelve, az akció minden rekord esetén lefut
- ha egy szabályhoz nincs akció hozzárendelve, az egész rekord kiírásra kerül

AWK minták

- elemi minták:
  - `(MINTA)`: csoportosítás
  - `!MINTA`: negáció
  - `/REGKIF/`: igaz, ha az egész rekord illeszkedik a reg.kif.-re
  - `KIF~/REGKIF/`: igaz, ha KIF illeszkedik REGKIF-re
  - `KIF!~/REGKIF/`: igaz, ha KIF nem illeszkedik REGKIF-re
- speciális minták:
  - `BEGIN`: csak a bemenet feldolgozása előtt teljesül
  - `END`: csak a kimenet feldolgozása után teljesül
- összetett minták:
  - `MINTA1&&MINTA2`: logikai ÉS
  - `MINTA1||MINTA2`: logikai VAGY

AWK konstansok

- numerikus konstansok
  - egész számok (12)
  - valós számok tizedesponttal (25.3)
  - egész vagy valós számok hatványkitevővel (1.234e+2)
- szöveges konstansok
  - `"STRING"`
  - `""` (üres string)
- reguláris kifejezések
  - `/REGKIF/`

AWK változók

- változók első használatkor automatikusan létrejönnek (nem kell őket deklarálni)
- változó típusa lehet numerikus, string, vagy tömb (később)
- számok és stringek közötti konverzió automatikus, de lehet manuálisan is:
  - stringből szám: adjunk hozzá 0-t
  - számból string: fűzzük hozzá az üres stringet
- értékadás egyszerűen `NÉV=ÉRTÉK`
- definiálatlan változók értéke típustól függően üres string vagy 0

AWK beépített változók

- `FILENAME`: aktuális bemeneti állomány neve
- `RS`: bemeneti rekordhatároló (default: sortörés)
- `FS`: bemeneti mezőhatároló (default: szóköz)
- `NR`: aktuális rekord sorszáma
- `FNR`: több fájl esetén aktuális rekord sorszáma az aktuális fájlon belül
- `NF`: aktuális rekord mezőinek száma
- `OFS`: kimeneti mezőhatároló
- `ORS`: kimeneti rekordhatároló
- `$0`: aktuális rekord teljes tartalma
- `$1, $2, $3`: aktuális rekord adott sorszámú mezőjének tartalma
- `$NF`: aktuális rekord utolsó mezőjének tartalma

AWK vezérlési szerkezetek

- használhatunk C-ben megszokott vezérlési szerkezeteket

```
if (PRED) {
        ...
} else {
        ...
}

while (PRED) {
        ...
}

do {
        ...
} while (PRED)
```