File searching in a directory hierarchy.

Department of Computer Systems FIT, Czech Technical University in Prague

```
find directory [expression]
```

- The find utility recursively descends the directory hierarchy for each path seeking files that match a Boolean expression.

- Parts of expression can be
  - grouped by `\(` and `\)`,
  - negated with `\!`,
  - separated by logical and `-a` (default) , logical or `-o`.

find

- **Actions**

`-print`	Always true. Causes the current pathname to be printed.
`-exec cmd {} \;`	True if the executed command returns a zero value as exit status.
`-ok cmd {} \;`	Like `-exec`, but it requires confirmation.
`-ls`	Always true. Prints current pathname together with statistics.

- **Predicates**

`-type [d,f,l,b,c]`	true if the type of the file is `d`, `f`, `l`, `b`, `c`
`-inum n`	true if the file has inode number *n*
`-name ´pattern´`	true if pattern matches the basename (use shell spec. character `*`, `?`, `[ ]`, …)
`-perm nnn`	true if file has given access permissions
`-mtime [n |-n |+n]`	true if the file's data was modified n days ago
`-atime [n |-n |+n]`	true if the file was accessed n days ago
`-ctime [n |-n |+n]`	true if the file's status was changed n days ago

Examples:

```
find $HOME
find $HOME -type f
find /bin -type f -name at
find /usr/bin -type f -name ´*grep´
find . -atime -1
find . -name ´[A-Z]*´
find . \! -name ´[A-Z]*´
```

Examples:

```
find . -name ´f*´ -print
find . -name ´s*´ -print
find . \( -name ´f*´ -o -name ´s*´ \) -print
find . \! \( -name ´f*´ -o -name ´s*´ \) -print

find . –print -name ´f*´ -print

find . –name core –ok rm { }  \;
find . –name core –exec rm { }  \;
```

xargs

- Construct argument lists and invoke utility.

- **Examples:**

```
mkdir ~/sources
find ~ -type f -name "*.c" | \
xargs -I { } -t mv { }  ~/sources/
```