

# Unix-like Operating Systems

Processes and jobs.

Jan Trdlička



Czech Technical University in Prague, Faculty of Information Technology  
Department of Computer Systems

# Contents

- 1 Discussion of homework
- 2 Processes – reports
- 3 Timing statistics
- 4 System – reports
- 5 Signals
- 6 Jobs
- 7 Homework

# Discussion of homework

- How to print names of regular files from your home directory, that have size bigger then 2 megabytes, and how to verify the result by the command `stat`?

# Discussion of homework

- How to print names of regular files from your home directory, that have size bigger then 2 megabytes, and how to verify the result by the command `stat`?

```
find ~ -type f -size +2M -exec stat --printf="%s %n\n" {} \;
```

# Discussion of homework

- How to print names of regular files from your home directory, that have size bigger then 2 megabytes, and how to verify the result by the command `stat`?

```
find ~ -type f -size +2M -exec stat --printf="%s %n\n" {} \;
```

- How to print names of all files from your home directory, that were accessed 7 days ago, and how to verify the result by the command `stat`?

# Discussion of homework

- How to print names of regular files from your home directory, that have size bigger then 2 megabytes, and how to verify the result by the command `stat`?

```
find ~ -type f -size +2M -exec stat --printf="%s %n\n" {} \;
```

- How to print names of all files from your home directory, that were accessed 7 days ago, and how to verify the result by the command `stat`?

```
find ~ -type f -atime +6 -exec stat --printf="%x\t%n\n" {} \;
```

# Discussion of homework

- How to print names of regular files from your home directory, that have size bigger than 2 megabytes, and how to verify the result by the command `stat`?

```
find ~ -type f -size +2M -exec stat --printf="%s %n\n" {} \;
```

- How to print names of all files from your home directory, that were accessed 7 days ago, and how to verify the result by the command `stat`?

```
find ~ -type f -atime +6 -exec stat --printf="%x\t%n\n" {} \;
```

- How to print names of regular files from the directory `/tmp`, that you can read, and how to verify the result by the command `ls`?

# Discussion of homework

- How to print names of regular files from your home directory, that have size bigger than 2 megabytes, and how to verify the result by the command `stat`?

```
find ~ -type f -size +2M -exec stat --printf="%s %n\n" {} \;
```

- How to print names of all files from your home directory, that were accessed 7 days ago, and how to verify the result by the command `stat`?

```
find ~ -type f -atime +6 -exec stat --printf="%x\t%n\n" {} \;
```

- How to print names of regular files from the directory `/tmp`, that you can read, and how to verify the result by the command `ls`?

```
find /tmp -type f \( \
    \( -user $USER -perm -400 \) \
    -o \( \(! -user $USER -group $(id -g) -perm -040 \) \
    -o \( \(! -user $USER \(! -group $(id -g) -perm -004 \) \) \
    -ls 2>/dev/null
```



# Discussion of homework

- How to print names of regular files from your home directory, that have size bigger than 2 megabytes, and how to verify the result by the command `stat`?

```
find ~ -type f -size +2M -exec stat --printf="%s %n\n" {} \;
```

- How to print names of all files from your home directory, that were accessed 7 days ago, and how to verify the result by the command `stat`?

```
find ~ -type f -atime +6 -exec stat --printf="%x\t%n\n" {} \;
```

- How to print names of regular files from the directory `/tmp`, that you can read, and how to verify the result by the command `ls`?

```
find /tmp -type f \( \
    \( -user $USER -perm -400 \) \
    -o \( \(! -user $USER -group $(id -g) -perm -040 \) \
    -o \( \(! -user $USER \(! -group $(id -g) -perm -004 \) \) \
    -ls 2>/dev/null
```

```
find /tmp -type f -readable -ls 2>/dev/null      # only GNU
```

# Processes – reports

- Run the command `passwd` in one terminal and solve the following question in the second terminal.
- How many processes has been started by the current user? Verify your answer.

# Processes – reports

- Run the command `passwd` in one terminal and solve the following question in the second terminal.
- How many processes has been started by the current user? Verify your answer.
  - Solution

```
ps -U $USER | tail -n +2 | wc -l
```

# Processes – reports

- Run the command `passwd` in one terminal and solve the following question in the second terminal.
- How many processes has been started by the current user? Verify your answer.
  - Solution

```
ps -U $USER | tail -n +2 | wc -l
```

```
pgrep -U $USER | wc -l
```

# Processes – reports

- Run the command `passwd` in one terminal and solve the following question in the second terminal.
- How many processes has been started by the current user? Verify your answer.
  - Solution

```
ps -U $USER | tail -n +2 | wc -l
```

```
pgrep -U $USER | wc -l
```

- Verification

```
ps -U $USER -o user=EFFECTIVE -o ruser=REAL \
-o pid,pcpu,comm
```

# Processes – reports

- Run the command `passwd` in one terminal and solve the following question in the second terminal.
- How many processes has been started by the current user? Verify your answer.
  - Solution

```
ps -U $USER | tail -n +2 | wc -l
```

```
pgrep -U $USER | wc -l
```

- Verification

```
ps -U $USER -o user=EFFECTIVE -o ruser=REAL \
-o pid,pcpu,comm
```

```
ps -U $USER -o user=EFFECTIVE -o ruser=REAL \
-o pid,pcpu,comm f # GNU
```

# Processes – reports

- Run the command `passwd` in one terminal and solve the following question in the second terminal.
- How many processes has been started by the current user? Verify your answer.
  - Solution

```
ps -U $USER | tail -n +2 | wc -l
```

```
pgrep -U $USER | wc -l
```

- Verification

```
ps -U $USER -o user=EFFECTIVE -o ruser=REAL \
-o pid,pcpu,comm
```

```
ps -U $USER -o user=EFFECTIVE -o ruser=REAL \
-o pid,pcpu,comm f # GNU
```

```
pgrep -lU $USER
```

# Processes – reports

- How many processes has been started by the user root?



# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

```
pgrep -U root | wc -l
```

# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

```
pgrep -U root | wc -l
```

- How many processes are running under the effective user identity of the user root?

# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

```
pgrep -U root | wc -l
```

- How many processes are running under the effective user identity of the user root?

```
ps -u root | tail -n +2 | wc -l
```

# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

```
pgrep -U root | wc -l
```

- How many processes are running under the effective user identity of the user root?

```
ps -u root | tail -n +2 | wc -l
```

```
pgrep -u root | wc -l
```

# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

```
pgrep -U root | wc -l
```

- How many processes are running under the effective user identity of the user root?

```
ps -u root | tail -n +2 | wc -l
```

```
pgrep -u root | wc -l
```

- How to print the processes which have different real and effective user identity?

# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

```
pgrep -U root | wc -l
```

- How many processes are running under the effective user identity of the user root?

```
ps -u root | tail -n +2 | wc -l
```

```
pgrep -u root | wc -l
```

- How to print the processes which have different real and effective user identity?

```
ps -e -o ruser,user,comm | \
  grep -v '^ *\[^[^ ]*\]' *1 '
```

# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

```
pgrep -U root | wc -l
```

- How many processes are running under the effective user identity of the user root?

```
ps -u root | tail -n +2 | wc -l
```

```
pgrep -u root | wc -l
```

- How to print the processes which have different real and effective user identity?

```
ps -e -o ruser,user,comm | \
  grep -v '^ *\[^[^ ]*\]' *1 '
```

```
ps -e -o ruser,user,comm | \
  awk '{ if ( $1 != $2 ) print $0 }'
```



# Processes – reports

- How many processes has been started by the user root?

```
ps -U root | tail -n +2 | wc -l
```

```
pgrep -U root | wc -l
```

- How many processes are running under the effective user identity of the user root?

```
ps -u root | tail -n +2 | wc -l
```

```
pgrep -u root | wc -l
```

- How to print the processes which have different real and effective user identity?

```
ps -e -o ruser,user,comm | \
  grep -v '^ *\[([^\]]*\] *)\[^\]]*\[^\]]*' *
```

```
ps -e -o ruser,user,comm | \
  awk '{ if ( $1 != $2 ) print $0 }'
```

```
ps -e -o ruser,user,comm | awk '$1 != $2'
```

# Timing statistics

- How to print timing statistics (real time, system time, user time,...) for the following processes?

- Commands

```
for (( i=0; i<100000; i++ )); do ;; done
```

```
du ~
```

```
sleep 5
```

```
ls; sleep 5;
```

- Solution

# Timing statistics

- How to print timing statistics (real time, system time, user time,...) for the following processes?

- Commands

```
for (( i=0; i<100000; i++ )); do ;; done
```

```
du ~
```

```
sleep 5
```

```
ls; sleep 5;
```

- Solution

```
time for (( i=0; i<100000; i++ )); do ;; done
```

# Timing statistics

- How to print timing statistics (real time, system time, user time,...) for the following processes?

- Commands

```
for (( i=0; i<100000; i++ )); do ;; done
```

```
du ~
```

```
sleep 5
```

```
ls; sleep 5;
```

- Solution

```
time for (( i=0; i<100000; i++ )); do ;; done
```

```
time du -s ~
```

# Timing statistics

- How to print timing statistics (real time, system time, user time,...) for the following processes?
  - Commands

```
for (( i=0; i<100000; i++ )); do ;; done
```

```
du ~
```

```
sleep 5
```

```
ls; sleep 5;
```

- Solution

```
time for (( i=0; i<100000; i++ )); do ;; done
```

```
time du -s ~
```

```
time sleep 5
```

# Timing statistics

- How to print timing statistics (real time, system time, user time,...) for the following processes?
  - Commands

```
for (( i=0; i<100000; i++ )); do ;; done
```

```
du ~
```

```
sleep 5
```

```
ls; sleep 5;
```

- Solution

```
time for (( i=0; i<100000; i++ )); do ;; done
```

```
time du -s ~
```

```
time sleep 5
```

```
time { du -s ~ ; sleep 5 ; }
```

# Timing statistics

- How to print timing statistics (real time, system time, user time,...) for the following processes?

- Commands

```
for (( i=0; i<100000; i++ )); do ;; done
```

```
du ~
```

```
sleep 5
```

```
ls; sleep 5;
```

- Solution

```
time for (( i=0; i<100000; i++ )); do ;; done
```

```
time du -s ~
```

```
time sleep 5
```

```
time { du -s ~ ; sleep 5 ; }
```

- Try to explain the previous results.

# System – reports

- How to print information about top cpu processes?



# System – reports

- How to print information about top cpu processes?

```
top
```

```
htop
```

```
prstat          # in Solaris
```

# System – reports

- How to print information about top cpu processes?

```
top
```

```
htop
```

```
prstat          # in Solaris
```

- Login to server `fray1.fit.cvut.cz`.
- How to print report information about processes and users by command `prstat`?

# System – reports

- How to print information about top cpu processes?

```
top
```

```
htop
```

```
prstat          # in Solaris
```

- Login to server `fray1.fit.cvut.cz`.
- How to print report information about processes and users by command `prstat`?

```
prstat -a       # in Solaris
```

# System – reports

- How to print information about top cpu processes?

```
top
```

```
htop
```

```
prstat          # in Solaris
```

- Login to server `fray1.fit.cvut.cz`.
- How to print report information about processes and users by command `prstat`?

```
prstat -a      # in Solaris
```

- How to print process tree?

# System – reports

- How to print information about top cpu processes?

```
top
```

```
htop
```

```
prstat          # in Solaris
```

- Login to server `fray1.fit.cvut.cz`.
- How to print report information about processes and users by command `prstat`?

```
prstat -a       # in Solaris
```

- How to print process tree?

```
pstree          # in Linux
```

```
ptree           # in Solaris
```

- How to print a list of available signals?

# Signals

- How to print a list of available signals?

```
kill -l
```

# Signals

- How to print a list of available signals?

```
kill -l
```

- Run the command `sleep 100 &`. How to send signal QUIT to this process?



# Signals

- How to print a list of available signals?

```
kill -l
```

- Run the command `sleep 100 &`. How to send signal QUIT to this process?

- `kill -QUIT $!`

# Signals

- How to print a list of available signals?

```
kill -l
```

- Run the command `sleep 100 &`. How to send signal QUIT to this process?

- `kill -QUIT $!`

- `kill -l quit`

```
kill -3 $!
```

# Signals

- How to print a list of available signals?

```
kill -l
```

- Run the command `sleep 100 &`. How to send signal QUIT to this process?

- `kill -QUIT $!`

- `kill -l quit`

```
kill -3 $!
```

- How to print a list of commands associated with each signal of the current shell?

# Signals

- How to print a list of available signals?

```
kill -l
```

- Run the command `sleep 100 &`. How to send signal QUIT to this process?

- `kill -QUIT $!`

- `kill -l quit`

```
kill -3 $!
```

- How to print a list of commands associated with each signal of the current shell?

```
trap
```

# Signals

- How to print a list of available signals?

```
kill -l
```

- Run the command `sleep 100 &`. How to send signal QUIT to this process?

- `kill -QUIT $!`

- `kill -l quit`

```
kill -3 $!
```

- How to print a list of commands associated with each signal of the current shell?

```
trap
```

- How to modify the behaviour of the current shell so that the shell will print "Hello", if it receives signal INT?

# Signals

- How to print a list of available signals?

```
kill -l
```

- Run the command `sleep 100 &`. How to send signal QUIT to this process?

- `kill -QUIT $!`

- `kill -l quit`

```
kill -3 $!
```

- How to print a list of commands associated with each signal of the current shell?

```
trap
```

- How to modify the behaviour of the current shell so that the shell will print "Hello", if it receives signal INT?

```
trap 'echo -e "\nHello"' INT
```

- How to reset the current shell to default behaviour for signal INT?

- How to reset the current shell to default behaviour for signal INT?

```
trap -- INT
```



- How to reset the current shell to default behaviour for signal INT?

```
trap -- INT
```

- How to run 100 times program `sleep 1000` in the background?

- How to reset the current shell to default behaviour for signal INT?

```
trap -- INT
```

- How to run 100 times program sleep 1000 in the background?

```
for ((i=0;i<100;i++)) ; do sleep 1000 & done
```

- How to reset the current shell to default behaviour for signal INT?

```
trap -- INT
```

- How to run 100 times program sleep 1000 in the background?

```
for ((i=0;i<100;i++)) ; do sleep 1000 & done
```

- How to kill all processes sleep started in the previous task?

- How to reset the current shell to default behaviour for signal INT?

```
trap -- INT
```

- How to run 100 times program sleep 1000 in the background?

```
for ((i=0;i<100;i++)) ; do sleep 1000 & done
```

- How to kill all processes sleep started in the previous task?

```
kill $(pgrep -U $USER -x sleep)
```

- How to reset the current shell to default behaviour for signal INT?

```
trap -- INT
```

- How to run 100 times program sleep 1000 in the background?

```
for ((i=0;i<100;i++)) ; do sleep 1000 & done
```

- How to kill all processes sleep started in the previous task?

```
kill $(pgrep -U $USER -x sleep)
```

```
pkill -U $USER -x sleep
```

- What is the difference between process ID (PID) and job ID (JID)?

- What is the difference between process ID (PID) and job ID (JID)?
- How to list all jobs in the current shell?

# Jobs

- What is the difference between process ID (PID) and job ID (JID)?
- How to list all jobs in the current shell?

```
jobs
```



# Jobs

- What is the difference between process ID (PID) and job ID (JID)?
- How to list all jobs in the current shell?

```
jobs
```

- How to run the command `sleep 1000` in foreground.

# Jobs

- What is the difference between process ID (PID) and job ID (JID)?
- How to list all jobs in the current shell?

```
jobs
```

- How to run the command `sleep 1000` in foreground.

```
sleep 1000
```

- What is the difference between process ID (PID) and job ID (JID)?
- How to list all jobs in the current shell?

```
jobs
```

- How to run the command `sleep 1000` in foreground.

```
sleep 1000
```

- How to move the previous command to the background without termination?

- What is the difference between process ID (PID) and job ID (JID)?
- How to list all jobs in the current shell?

```
jobs
```

- How to run the command `sleep 1000` in foreground.

```
sleep 1000
```

- How to move the previous command to the background without termination?

```
CRTL Z  
bg
```

- What is the difference between process ID (PID) and job ID (JID)?
- How to list all jobs in the current shell?

```
jobs
```

- How to run the command `sleep 1000` in foreground.

```
sleep 1000
```

- How to move the previous command to the background without termination?

```
CRTL Z  
bg
```

- How to terminate the previous command?

- What is the difference between process ID (PID) and job ID (JID)?
- How to list all jobs in the current shell?

```
jobs
```

- How to run the command `sleep 1000` in foreground.

```
sleep 1000
```

- How to move the previous command to the background without termination?

```
CRTL Z  
bg
```

- How to terminate the previous command?

```
kill %1
```

- How to print a name of the process which occupies the most physical memory?
- How to print a name of the process which has the most threads?
- Login to the server `fray1.fit.cvut.cz`.  
How to print a name of the process which spent the most time on CPU (cumulative time)?