

```
1: BEGIN { s = 0 ; OFS=" " }
2:       { if ($3==name) { printf "%.1fMB ", $5/1024/1024
3:                               print $6,$7,$8,$9
4:                               s += $5 }
5:       }
6: END { if (s>0) {print; print; print "Gesamt:",s/1024/1024/1024, "GByte"} }
```

```
1: #!/bin/sh
2: #
3: #  Convert C++ source files to pdf
4: #*****
5:
6: #SOURCE_ORDER='ls *.cpp *.cu *.h* *.tpp *.tcc 2>/dev/null '
7: #SOURCE_ORDER='find -regex '.*\.(cpp|h|hpp|cu|tpp|tcc)' 2>/dev/null '
8: SOURCE_ORDER='find -regex '.*' 2>/dev/null '
9: LECTURE='basename ${PWD}'
10:
11: echo ${SOURCE_ORDER}
12: echo ${LECTURE}
13:
14: #enscript ${SOURCE_ORDER} -o ${LECTURE}.ps --color=1 -Ecpp -C -fCourier8
15: enscript -r ${SOURCE_ORDER} -o ${LECTURE}.ps --color=1 -Ebash -C -fCourier10
16: ps2pdf ${LECTURE}.ps
17: rm ${LECTURE}.ps
18:
19: # enscript --help-highlight
20:
```

```
1: #!/bin/bash
2: # http://wiki.ubuntuusers.de/Shell/Bash-Skripting-Guide\_f%C3%BCr\_Anf%C3%A4nger
3: #http://wiki.bash-hackers.org/commands/classicstest
4:
5: #####
6: #      Anfang der Funktion
7: #####
8: function punkte_pro_student ()
9: {
10:     echo " "
11:     echo " "
12:     echo "#####"
13:     echo "### " $i "###" # switch to sub dir
14:     echo "#####"
15:     pushd $i > /dev/null
16:     #echo "---> " $PWD
17:
18:     FILES=`find . -name *.cpp`
19:     #echo ${FILES}
20:
21:     for s in ${FILES}
22:     do
23:         echo " "; echo "##### " $s " #####"; echo " "
24:         echo "GCCCHECK:" g++ -c $CXXFLAGS $s
25:         g++ -c $CXXFLAGS $s || echo "***** ERROR *****"
26:         echo " "
27:         echo "CPPCHECK:" cppcheck $CHECKFLAGS $s
28:         cppcheck $CHECKFLAGS $s
29:         echo " "
30:         echo "CLANG CHECK:" clang++ -c $CLANGFLAGS $s
31:         clang++ -c $CLANGFLAGS $s
32:         echo " "
33:     done
34:
35:
36:     popd > /dev/null
37:     return
38: }
39: #####
40: #      Ende der Funktion punkte_pro_student ()
41: #####
42:
43: #####
44:
45: MY_BINDIR=${PWD}/../my_bin
46:
47: rm -rf log
48: mkdir log
49:
```

```
50:
51:
52: #CXXFLAGS=" -m32 -pedantic -std=c++11 -Weffc++ -Wall -Wextra -pedantic -Wswitch-default -Wfloat-equal -Wundef -Wredundant-decls -Winit-self -Wshadow -Wparentheses -Wshadow -Wunreachable-code -Wuninitialized -Wmaybe-uninitialized"
53: CXXFLAGS="-DNDEBUG -pedantic -std=c++11 -Weffc++ -Wall -Wextra -pedantic -Wno-sign-compare -Wswitch-default -Wfloat-equal -Wundef -Wredundant-decls -Winit-self -Wshadow -Wparentheses -Wshadow -Wunreachable-code -Wuninitialized -Wmaybe-uninitialized"
54: # -Wmissing-declarations
55: CHECKFLAGS="-DNDEBUG --enable=all --inconclusive --std=c++11 --std=posix --suppress=missingIncludeSystem"
56: #CLANGFLAGS="-fsyntax-only -Weverything -pedantic"
57: #CLANGFLAGS="-std=c++11 -fsyntax-only -Wdocumentation -Wconversion -Wshadow -Wfloat-conversion -pedantic"
58: CLANGFLAGS="-DNDEBUG -std=c++11 -fsyntax-only -Wdocumentation -Wconversion -Wshadow -pedantic -Wno-sign-conversion"
59:
60:
61: rm -rf log 2>/dev/null
62:
63: DIRS=`ls -d */ | tr -d / `          # Verzeichnisse
64: echo $DIRS
65:
66: mkdir log
67:
68:
69: for i in $DIRS
70: do
71:     echo "***** " ${i} " *****"
72:     COMPILE_LOG=log/${i}_compile.log
73:     rm -f $COMPILE_LOG >/dev/null
74:     echo " " > $COMPILE_LOG
75:
76: #     punkte_pro_student &> $COMPILE_LOG
77: #punkte_pro_student 2>&1 | tee -a $COMPILE_LOG
78: punkte_pro_student &>> $COMPILE_LOG
79: #punkte_pro_student
80: done
81: find . -name "*.o" -exec \rm -f {} \;
82: find . -name "a.out" -exec \rm -f {} \;
83:
84:
85: ##### kopiere alle ccompile-logs in ein File ###
86: #GLOBAL_LOG=log/_gcompile.log
87: #cat log/*_compile.log > $GLOBAL_LOG
88:
89: ##### Pruefe auf kopierte Loesungen #####
90: #CHECK_LOG=log/_check4copy.log
91: #check4copy > $CHECK_LOG
92:
93: unix2dos log/*
94:
95: # https://www.cyberciti.biz/faq/grep-regular-expressions/
```

```
96: # https://www.regular-expressions.info/refquick.html
97: # Check for pointer
98: #grep -E "(int|float|double|char|bool) *\*" main.cpp
99: #grep -ER "(int|float|double|char|bool) *\*" . |grep .cpp |grep -v log |grep -v html
100: #grep -ER "(int|float|double|char|bool) *\*" . |grep .cpp |grep -E -v '(log|html)'
101: #grep -ER "(int|float|double|char|bool) *\*[a-Z ]" . |grep .cpp |grep -E -v '(log|html)'
102:
103: #find . -name "*.h" -o -name "*.cpp" -exec grep -E "(int|float|double|char|bool) *\*" {} \;
104:
105: # check for small files (Betrug !?)
106: # ls -ls */*/*.cpp
107:
108:
109: # https://www.cyberciti.biz/faq/grep-regular-expressions/
110: #
111: # https://www.gnu.org/software/sed/manual/html\_node/Regular-Expressions.html
112: #sed s/"switch(.*)"//g Switch.cpp
113: # https://www.gnu.org/software/sed/manual/html\_node/Text-search-across-multiple-lines.html
114: #sed s/"switch.*{\{1\}.*[c].*break.*}\{1\}"//g < a.txt
115:
```

./konvert.sed

Mon Mar 08 18:10:05 2021

1

1: s/\\tilde/\\widetilde/g

```
1: #!/bin/sh
2: # Find all files larger than 500 kByte and print their names and sizes.
3: # These files have not been modified during the last 30 days
4: #     or have not been accessed the last 120 days.
5: # Send email to owners of files.
6:
7: AWKFILE1=${HOME}/bin/Large.awk
8: FILE=gefunden
9:
10: NAMELIST=ghaase
11: # "braeuer cazier creator ghaase guest hendorf heise joachim jung kick kuhn ulanger"
12: rm -f *.lrg ${FILE}
13:
14: find . -size +4M \( -mtime +30 -o -atime +120 \) -type f -exec ls -l {} \; > ${FILE}
15: # find ${HOME} -size +1M \( -mtime +30 -o -atime +120 \) -type f -exec ls -l {} \; > ${FILE}
16:
17: for NAME in ${NAMELIST}
18: do LRG=${NAME}.lrg
19:     if [ ! -s ${LRG} ]
20:     then
21:         echo "Files larger than 1MByte that have not been accessed a long period " > ${LRG}
22:         echo "" >> ${LRG}
23:         awk -f ${AWKFILE1} name=$NAME $FILE >> ${LRG}
24:         #     mail ${NAME} < $LRG
25:     fi
26: done
27:
28: exit 0
```

```
1: #!/bin/bash
2:
3: #      Synchronizes the recent directory on meth047211 with the same directorty on faust
4: #
5:
6: MASTER=math047056
7: #SLAVE=ghaase@143.50.47.46
8: SLAVE=ghaase@143.50.47.211
9:
10:
11: #INCLUDE_FILE=bin/include.rsync
12: EXCLUDE_FILE=~/.bin/exclude.rsync
13:
14:
15: # RSYNC_FLAGS="-auRz --cvs-exclude --exclude-from=$EXCLUDE_FILE"
16: RSYNC_FLAGS="-auvzR --cvs-exclude --exclude-from=$EXCLUDE_FILE"
17:
18: # remove Home directory from path of recent directory
19: DIR=`pwd -L`
20: DIR=${DIR#*ghaase/}
21: # echo $DIR
22:
23: echo "Synchronize " $DIR " from " $MASTER " to " $SLAVE
24: rsync $RSYNC_FLAGS . $SLAVE:$DIR
25: # echo "Synchronize " $DIR " from " $SLAVE " to " $MASTER
26: # rsync $RSYNC_FLAGS $SLAVE:$DIR ~
27:
28:
29:
30:
31:
32:
```



```
1: #!/bin/sh
2: #
3: #  Convert C++ source files to pdf
4: #*****
5:
6: #SOURCE_ORDER='ls *.cpp *.cu *.h* *.tpp *.tcc 2>/dev/null `
7: SOURCE_ORDER='find -regex '.*\.\.(cpp|h|hpp|cu|tpp|tcc)` 2>/dev/null `
8: LECTURE='basename ${PWD} `
9:
10: echo ${SOURCE_ORDER}
11: echo ${LECTURE}
12:
13: #enscript ${SOURCE_ORDER} -o ${LECTURE}.ps --color=1 -Ecpp -C -fCourier8
14: enscript -r ${SOURCE_ORDER} -o ${LECTURE}.ps --color=1 -Ecpp -C -fCourier10
15: ps2pdf ${LECTURE}.ps
16: rm ${LECTURE}.ps
17:
18: # enscript --help-highlight
19:
```

```
1: #!/bin/sh
2: # see https://wiki.gnome.org/Apps/Gedit/ExternalToolsPluginCommands
3:
4: # see astyle: http://astyle.sourceforge.net/astyle.html
5: # http://astyle.sourceforge.net/astyle.html#\_Brace\_Style\_Options
6:
7: # astyle --style=kr --indent=tab=4 --indent-switches --indent-preprocessor --pad-oper --pad-header --break-closing-brac
kets --align-pointer=name --lineend=linux $GEDIT_CURRENT_DOCUMENT_NAME
8:
9: #astyle --style=stroustrup --indent=spaces=4 --indent-switches --indent-modifiers --indent-preprocessor --pad-oper --pa
d-header --break-closing-brackets --align-pointer=name --lineend=linux $1
10:
11: astyle --style=stroustrup --indent=spaces=4 --indent-switches --indent-preprocessor --pad-oper --pad-header --break-clo
sing-brackets --align-pointer=name --lineend=linux --close-templates --align-reference=name --keep-one-line-blocks --preserve-
date $1
12:
13: # --pad-comma      (ab astyle 3.1)
14:
15: #astyle --style=allman --indent=spaces=4 --indent-switches --indent-preprocessor --pad-oper --pad-header --break-closin
g-brackets --align-pointer=name --lineend=linux $1
16:
```

```
1: #!/bin/sh
2: #
3: # Konvertierung von Zeichenketten, rekursiv ab dem aktuellen Verzeichnis
4: #*****
5:
6: OLDSTRING=www.uni-graz.at/~haasegu
7: NEWSTRING=imsc.uni-graz.at/haasegu
8:
9: FILETYPES="\( -name *.tex -o -name *.html -o -name *.cpp -o -name *.c -o -name *.h -o -name *akefile -o -name *
default \)"
10:
11:
12: echo
13: echo '      ' ${OLDSTRING} '      -----> ' ${NEWSTRING}
14: echo
15:
16: #find . \( -name *.tex -o -name *.html -o -name *.cpp -o -name *.c -o -name *.h -o -name *akefile -o -name *de
fault \) \
17:      #-ok sed -i -e's/${OLDSTRING}/${NEWSTRING}/g' {} \;
18:
19: find . \( -name *.tex -o -name *.html -o -name *.cpp -o -name *.c -o -name *.h -o -name *akefile -o -name *def
ault \) -type f \
20:      -exec sed -i -e's/www.uni-graz.at/%7Ehaasegu/imsc.uni-graz.at/haasegu/g' {} \;
21:
22:
```

```
1: #!/bin/sh
2: #
3: #   Konvertierung von Zeichenketten
4: #*****
5:
6: echo
7: echo '      [Pp]rojekt  ----->  [Pp]roject'
8: echo
9:
10: if [ $# -eq 0 ]
11: then echo "      Aufruf :   konvert filename[n]"
12:      echo
13:      exit 1
14: fi
15:
16: for i in `ls $@`
17: do echo " " >> $i
18:     sed -e "s/A{/Obj{/g" $i >$i.temp
19: #       sed -f ./konvert.sed $i >$i.temp
20:     echo " ==> "$i
21:     if [ -s $i.temp ]
22:     then mv $i.temp $i
23:     fi
24: done
25:
26: echo
27: exit 0
28:
29: #in-place
30: # sed -i -e's/last entry/first entry/g' getmatrix.h
31:
32: # for i in `grep -l "last entry" */*/*.h`; do sed -i -e's/last entry/first entry/g' $i; done
33:
```

```
1: #!/bin/sh
2: # Find all files hat are not needed and removes them
3: # These files have not been modified during the last 30 days
4: #     or have not been accessed the last 120 days.
5: # Send email to owners of files.
6:
7:
8: #find ${HOME} \( -name '*.log' -o -name '*.aux' -o -name '*.blg' \
9: #               -o -name 'core*' -o -name '*.bak' -o -name '.*~' \) \
10: #               \( -type f -o -mtime +3 -o -atime +3 \) -exec ls -l {} \; > auxclean.txt
11:
12: # find ${HOME} \( -name '*.log' -o -name '*.aux' -o -name '*.blg'
13: find . \( -name '*.log' -o -name '*.aux' -o -name '*.blg' \
14:          -o -name 'core' -o -name '*.bak' -o -name '*~' \
15:          -o -name '*.o' -o -name '*.exe' -name 'a.out' -o -name '*.asv' \
16:          -o -name '*.backup' -o -name '*.dvi' \) \
17:          \( -type f -o -mtime +3 -o -atime +3 \) -print -exec \rm -f '{}' \;
18:
19: exit 0
20:
21:
22: find /tmp -ctime +2 -exec rm -r -f '{}' \;
23: find /usr/tmp -ctime +2 -exec rm -r -f '{}' \;
24:
```

```
1: #!/bin/bash
2: # Rescale pictures given as parmeter
3:
4: #                                decrease width and height by 25%
5: OPT_CONVERT="-resize 25%"
6: #OPT_CONVERT="-resize 10%"
7: #OPT_CONVERT="-resize 55%"
8:
9: #                                standard prefix
10: #STD_PREFIX=small/med_
11: STD_PREFIX=med_
12:
13: if [ $# -lt 1 ]
14: then
15:     echo " "
16:     echo "$0 converts pictures to a different size."
17:     echo " "
18:     echo "Correct call:  $0 \"filename\" [prefix_for_files]"
19:     echo "          e.g.:  $0 \"h*.jpg\"    small"
20:     echo " "
21:     exit 9
22: fi
23:
24: if [ $# -gt 2 ]
25: then
26:     echo " "
27:     echo "To much arguments !!!"
28:     echo "You have to use \"\" around the option filename."
29:     echo " "
30:     echo "Correct call:  $0 \"filename\" [prefix_for_files]"
31:     echo "          e.g.:  $0 \"h*.jpg\"    small"
32:     echo " "
33:     exit 9
34: fi
35:
36: if [ $# -lt 2 ]
37: then
38:     echo "Missing second parameter."
39:     echo "Using standard prefix  " ${STD_PREFIX} "  instead"
40:     echo ""
41:     PREFIX=${STD_PREFIX}
42: else
43:     PREFIX="$2_"
44: fi
45:
46: echo "Conversion uses prefix  " ${PREFIX}; echo ""
47:
48: FILES=`ls $1`
49:
```

```
50: for NAME in ${FILES}
51: #for NAME in $*
52: do
53:     NEWNAME=${PREFIX}${NAME}
54:     echo "${NAME} --> ${NEWNAME}"
55:     convert ${OPT_CONVERT} ${NAME} ${NEWNAME}
56: done
57:
58: exit 0
```