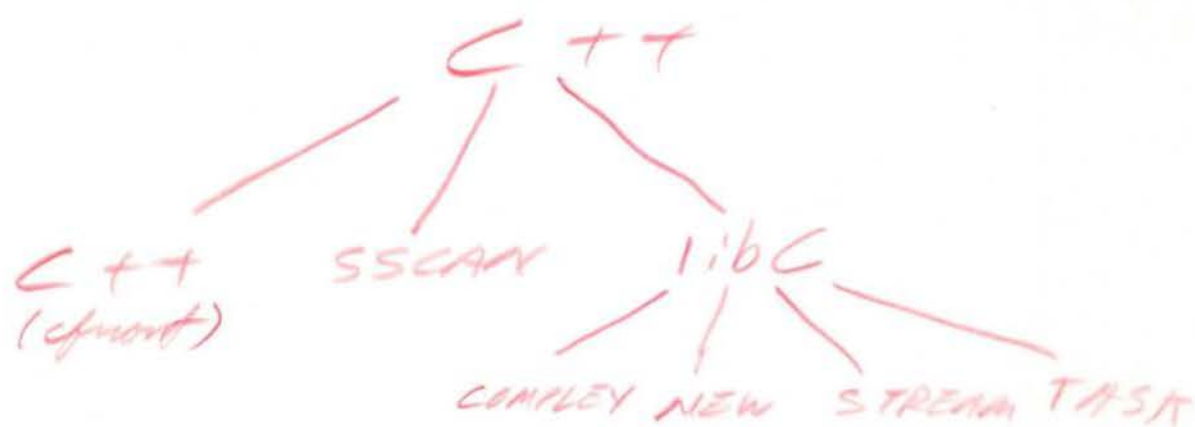




Bye,

Here's a copy of what
went out ~~with~~ on the tape.
Sorry about the delay, but
you know how it is when you
have source control and multiple
versions to mess things up!

SCP

XX
XX
XX
XX

795 - laser1

Sat Feb 9 19:32:27 EST 1985

ARGS: [-C -d -fp -oorig=c++ -otime=476843513]
NAME: [c++]
PASSCOMM: [sfbc!c++:D.Swartout:lpbin=m13:home=bc]
copies: [1]
ctime: [0]
files: [/usr/spool/lp/request/laser/d0-9677]
job: [laser-9001]
lpbin: []
mach: [sfbc (laser-9001)]
mail: []
originator: [c++]
printer: [laser1]
printtime: [476843537]
request: [laser-9677]
route: [nsc]
sequence: [795]
subtime: [476843513]
title: []
transit: [0]
version: [08-02-84]

%Z% %M% %I% %H% %T%

To build and install your C++ compiler:

1) Dump the tape (using "cpio -icd") to a convenient directory (like "/home/directory/C++"). No doubt this has already been done.

2) Edit the file localCC. Change the ROOT variable to /home/directory/C++.

3) cd to the directory and issue
"make ROOT=/home/directory/C++ fromscratch"
This will make everything from scratch and install the executables, libraries and include files locally (i.e., under /home/directory/C++).
Issue a "make clean".

4) Issue (from /home/directory/C++)
"make CC=/home/directory/C++/usr/lbin/CC install"
This will install the executables, libraries and include files in standard places:
/usr/lbin/cfront
/usr/lbin/CC
/lib/libC.a
/usr/include/CC/...
/usr/lbin/sscan
/lib/CCcrt0.o
/lib/CCmcr0.o

5) To save space, issue a "make clean".

```
# %Z% %M% %I% %H% %T%  
# C++ master makefile  
#
```

```
ROOT=  
CC=CC  
CCLOC=/usr/lbin/CC  
CFRONT=/usr/lbin/cfront  
LIBC=/lib/libC.a  
SSCAN=/usr/lbin/sscan  
INCL=/usr/include/CC  
CRTLIB=/lib  
LSRC=/usr/lsrc/cmd/CC
```

```
all      :  
    if u3b; then \  
        cd C++ ; make CC=$(CC) OSUF=.o ; \  
        cd .. ; \  
        cd libC ; make CC=$(CC) OSUF=.o ; \  
    else \  
        cd C++ ; make CC=$(CC) OSUF=.o ; \  
        cd .. ; \  
        cd libC ; make CC=$(CC) OSUF=.o ; \  
    fi  
    if u3b ; \  
        then cd sscan ; make CC=$(CC) ; \  
    fi
```

```
install :  
    if u3b; then \  
        cd C++ ; make CC=$(CC) OSUF=.o ; \  
    else \  
        cd C++ ; make CC=$(CC) OSUF=.o ; \  
    fi  
    if u3b ; \  
        then cd sscan ; make CC=$(CC) ; \  
    fi  
    rm -f $(ROOT)$(CCLOC)  
    if test $(ROOT) ; then \  
        cp localCC $(ROOT)$(CCLOC) ; \  
    else \  
        cp CC $(CCLOC) ; \  
    fi  
    chmod +x $(ROOT)$(CCLOC)  
    rm -f $(ROOT)$(CFRONT)  
    cp C++/cfront $(ROOT)$(CFRONT)  
    if u3b; then \  
        rm -f $(ROOT)$(CRTLIB)/CCert0.o ; \  
        cp sscan/CCert0.o $(ROOT)$(CRTLIB)/CCert0.o ; \  
        rm -f $(ROOT)$(CRTLIB)/CCmcert0.o ; \  
        cp sscan/CCmcert0.o $(ROOT)$(CRTLIB)/CCmcert0.o ; \  
        rm -f $(ROOT)$(SSCAN) ; \  
        cp sscan/sscan $(ROOT)$(SSCAN) ; \  
    fi  
    if u3b; then \  
        cd libC ; make CC=$(CC) OSUF=.o ; \  
    else \  
        cd libC ; make CC=$(CC) OSUF=.o ; \  
    fi
```

```
        cd libC ; make CC=$(CC) OSUF=..o ; \  
fi  
rm -f $(ROOT)$(LIBC)  
cp libC$(CRTLIB)C.a $(ROOT)$(LIBC)  
rm -rf $(ROOT)$(INCL)/*  
cp incl/sys/* $(ROOT)$(INCL)/sys  
cp incl/*.h $(ROOT)$(INCL)
```

```
clean  :  
  
        cd C++ ; make clean  
        cd sscan ; make clean  
        cd libC ; make clean  
        cd c ; rm -f *.o
```

```
clobber :  
  
        cd C++ ; make clobber  
        cd sscan ; make clobber  
        cd libC ; make clobber
```

```
uplist  :  
  
        uplist -f up.C++.3bs  
        rm -f $(LSRC)/C++/*  
        cp C++/*.[chy] C++/makefile $(LSRC)/C++  
        rm -f $(LSRC)/sscan/*  
        cp sscan/*.[cs] sscan/makefile $(LSRC)/sscan  
        rm -f $(LSRC)/incl/*.h $(LSRC)/incl/sys/*  
        cp incl/*.h $(LSRC)/incl  
        cp incl/sys/* $(LSRC)/incl/sys  
        rm -f $(LSRC)/libC/makefile  
        cp libC/makefile $(LSRC)/libC  
        rm -f $(LSRC)/libC/complex/*  
        cp libC/complex/*.[ch] libC/complex/makefile $(LSRC)/libC/complex  
        rm -f $(LSRC)/libC/new/*  
        cp libC/new/*.[ch] libC/new/makefile $(LSRC)/libC/new  
        rm -f $(LSRC)/libC/stream/*  
        cp libC/stream/*.[ch] libC/stream/makefile $(LSRC)/libC/stream  
        rm -f $(LSRC)/libC/task/*  
        cp libC/task/*.[chs] libC/task/makefile $(LSRC)/libC/task  
        rm -f $(LSRC)/CC $(LSRC)/makefile  
        cp CC makefile $(LSRC)  
        uplist -f up.C++.vaxes
```

```
fromscratch :  
        cd $(ROOT)/c ; cc *.c -o cfront  
        mv $(ROOT)/c/cfront $(ROOT)/usr/lbin/cfront  
        cp $(ROOT)/localCC $(ROOT)/usr/lbin/CC  
        chmod +x $(ROOT)/usr/lbin/CC  
        if u3b; then \  
                cd libC/new ; make CC=$(ROOT)/usr/lbin/CC OSUF=..o ; \  
                ar cr libC.a $(ROOT)/libC/new/*.o ; \  
                mv libC.a $(ROOT)/lib ; \  
                cd ../.. ; \  
                make CC=$(ROOT)/usr/lbin/CC ROOT=$(ROOT) OSUF=..o install ;  
        else \  
                cd libC/new ; make CC=$(ROOT)/usr/lbin/CC OSUF=..o ; \  
                ar cr libC.a $(ROOT)/libC/new/*.o ; \  
        fi
```

Feb 8 14:11 1985 makefile Page 3

```
        mv libC.a $(ROOT)/lib ; \  
        cd ../../ ; \  
        make CC=$(ROOT)/usr/lbin/CC ROOT=$(ROOT) OSUF=..o install ;  
fi  
cp $(ROOT)/localCC $(ROOT)/usr/lbin/CC  
chmod +x $(ROOT)/usr/lbin/CC
```

@(#) CC 1.2 2/8/85 17:59:07

E=0

: \${CFRONT:=\$ROOT/usr/lib/cfront}

: \${SSCAN:=\$ROOT/usr/lib/sscan}

: \${ARLIB:=\$ROOT/lib/libC.a}

: \${I:=\$ROOT/usr/include/CC}

: \${CCOMP:=cc}

: \${LDCC:=/usr/local/bin/cc}

: \${CRTOPT:='-Y S,\$ROOT/lib/CC'}

if vax ; then

SUF=..c

OSUF=..o

else

SUF=.i

OSUF=.o

fi

for A do

case \$A in

+V) I="/usr/include"

O="\$O \$A"

;;

+i) RMI="echo preprocessor output in"

;;

+I) STATICS=yes ;

O="\$O \$A"

;;

+*) O="\$O \$A"

;;

-E) PON=1

;;

-F) FON=1

;;

-Fc) FON=1

O="\$O +L"

;;

-c) DASHCOPT="-c"

;;

-r) LDOPT="\$LDOPT \$A"

DASHROPT="-r"

;;

-[gp]) COMPOPT="\$COMPOPT \$A"

LDOPT="\$LDOPT \$A"

;;

-C) Y="\$Y \$A"

O="\$O +C"

;;

-I*) Y="\$Y \$A"

;;

-D*) Y="\$Y \$A"

;;

-U*) Y="\$Y \$A"

;;

-.c) echo "bad suffix option: -.c"

;;

```

-.*)    SUF=`expr "$A" : '-\(.*)'\`
        ;;
[!-]*.[oa])    LDOPT="$LDOPT $A"
        G=1
        ;;
*.c)    if    test -f $A
        then
            B=`basename $A .c`
            if    test $PON
            then
                if test $SUF
                then
                    echo "cpp $O $A > $B$SUF:"
                    /lib/cpp $Y -I$I $A >$B$SUF
                    E=$?
                    case $E in
                        0)    ;;
                        *)    break
                               ;;
                    esac
                else
                    /lib/cpp $Y -I$I $A
                    E=$?
                    case $E in
                        0)    ;;
                        *)    break
                               ;;
                    esac
                fi
            fi
        elif
        then
            if test $SUF
            then
                echo "cfront $O < $A > $B$SUF:"
                /lib/cpp -C $Y -I$I $A | $CFRONT +f$A $O
                E=$?
                if [ "$RMI" != "" ] ; then
                    sed '/^#/d' >/tmp/CC$$ $B$SUF
                    mv /tmp/CC$$ $B$SUF
                fi

                case $E in
                    0)    ;;
                    1)    echo "1 error"
                           break
                           ;;
                    *)    echo "$E errors"
                           break
                           ;;
                esac
            else
                /lib/cpp -C $Y -I$I $A | $CFRONT +f$A $O
                E=$?
                case $E in
                    0)    ;;
                    *)    break
                           ;;
                esac
            fi
        fi
    fi

```

```

;;
esac
fi
else
echo "CC $O $A:"
/lib/cpp -C $Y -I$I $A | $CFRONT +f$A $O > $B$SUF
E=$?
if [ "$RMI" != "" ] ; then
sed '/^#/d' >/tmp/CC$$ $B$SUF
mv /tmp/CC$$ $B$SUF
fi

R=1
C="$C $B$SUF"
case $E in
0) COMPOPT="$COMPOPT $B$SUF"
LDOPT="$LDOPT $B$SUF"
G=1
;;
1) echo "1 error"
break
;;
*) echo "$E errors"
break
;;
esac
fi
else
echo "$A not found"
fi
*) COMPOPT="$COMPOPT $A"
LDOPT="$LDOPT $A"
G=1
;;
esac
done
if
then
test $G
$CCOMP -c $COMPOPT
if [ "$DASHCOPT" = "-c" ]
then
: no ld
else
if [ "$STATICS" = "yes" -a "$DASHROPT" != "-r" ]
then
$SSCAN $LDOPT >/tmp/_ST$$.$c
$LDCC $CRTOPT /tmp/_ST$$.$c $LDOPT $ARLIB
rm -f /tmp/_ST$$.$c _ST$$.$o _ST$$.$s 2>/dev/null
else
$LDCC $LDOPT $ARLIB
fi
fi
fi

```

Feb 8 17:59 1985 CC Page 4

```
if      test $R
then    ${RMI:-rm} $C
fi
exit $E
```