Dokumentacja przeniesienia oprogramowania ANSA 4.0 na komputer Convex 3200

Jerzy Pawlus

Akademickie Centrum Komputerowe
CYFRONET

Luty 1993

Spis treści

1 **2**
1.1 Wprowadznie 2
1.2 Opis plików konfiguracyjnych ANSA dla systemu ConvexOS 10.0 2

2 **12**
2.1 Przygotowanie środowiska do kompilacji systemu ANSA 12

3 **16**
3.1 Kompilacja systemu ANSA 16
3.2 Sprawdzenie poprawności instalacji systemu ANSA 26

Rozdział 1

1.1 Wprowadznie

Niniejszy raport jest wynikiem prac związanych z zainstalowaniem oprogramowania ANSA na komputerze Convex 3200 pracującym pod kontrolą systemu operacyjnego ConvexOS 10.0. Prace instalacyjne były prowadzone zgodnie ze wskazówkami zawartymi w rozdziale 11 podręcznika **ANSA System Manager's Guide**.
Oprócz normalnych trudności związanych z utworzeniem odpowiednich plików konfiguracyjnych dla nowej platformy sprzętowej pojawiły się nowe, związane ze specyficznymi własnościami systemu operacyjnego ConvexOS 10.0. Wszystkie powstałe w ten sposób problemy zostały udokumentowane a odpowiednie poprawki do plików źródłowych systemu ANSA przedstawione jako wynik działania programu **diff** na wersji oryginalnej i zmodyfikowanej.

1.2 Opis plików konfiguracyjnych ANSA dla systemu ConvexOS 10.0

Pierwowzorem plików konfiguracyjnych systemu ANSA dla systemu ConvexOS 10.0 były pliki dla systemu SUN OS 4.1. Było to umotywowane wspólnym pochodzeniem obu systemów, wywodzącymi się z systemu UNIX 4.3BSD oraz faktem znajomości systemu SUN OS 4.1 przez przenoszącego.

Pliki konfiguracyjne zostały przedstawione w całości. Jedynym wyjątkiem są pliki **stack.cpp** i **ACTION**. Zamieszczone fragmenty tych plików zawierają wyłącznie kod dopisany dla komputera Convex.

- **ANSAware/ACTION**
  Do pliku ACTION dołączono wiersz opisujący nową platformę sprzętową. Poniżej opisano poszczególne pola tego wiersza:

  ansa/ansaware/ACTION

Pole 1:	convex3200_cos_10.0	
Pole 2:	unix	
Pole 3:	srl	
Pole 4:	config.h	(capsule/config.h)
Pole 5:	convex.h	(capsule/opsys.h)
Pole 6:	convex.h	(capsule/machine.h)
Pole 7:	convex3200_cos.h	(capsule/stack.h)
Pole 8:	convexIEEE.h	(stub/moveMacs.h)
Pole 9:	unix	
Pole 10:	unix	
Pole 11:	ConvexOS.10.0 on a Convex 3200	

- **include/capsule/stack**
  Opisywana instalacja oprogramowania ANSA na komputerze Convex nie wykorzystuje standardowych procedur Unixa **setjmp** i **longjmp** do przełączania stosu. Powodem jest fakt sprawdzenia przez procedure **longjmp** czy powrót następuje do środowiska zachowanego przez procedurę **setjmp**.

  ansa/gen_dist/include/capsule/stack/convex3200_cos.h

```
/* Header Start */
/*
 *  Copyright 1989,1990,1991 APM Limited (ISA Project)
 *
 *  File: sun3_sos.h
 *  Author:
 *  Date Created:
 *
 *  General Description:
 *    Interface to stack handler for a Sun3 running SunOS.
 *
 *  $Header: /disc2/usr/local/src/ANSAware4.0/master/include/capsule/stack/RCS/
sun3_sos.h,v 1.5 1992/02/08 17:53:38 can Exp $
 *
 */
/* Header End */

#ifndef __STACK_INCLUDED__
#define __STACK_INCLUDED__

#include "options.h"

/*
 * This file sets up constants to allow multi-tasking.  This requires the
 * infrastructure associating a stack and register dump with each task -
 * actually a task IS the stack and register dump.
 *
```

```
 * We assume a mechanism like Unix setjmp/longjmp where an array (called
 * a jmp_buf) holds all volatile state (i.e. PSW and registers) associated
 * with an executing task.  Within this array, two fields describing the
 * current stack frame are of interest:
 *
 * STACK_FRAME (often called the Frame Pointer or FP) points to the
 * beginning of the current stack frame,
 *
 * STACK_FRONT (often called the Stack Pointer or SP) points to the end of
 * the current stack frame, if another call is made then this
 * value is copied into STACK_FRAME.
 *
 * these fields may be found in jmp_buf[STACK_FRAME] and jmp_buf[STACK_FRONT]
 *
 * For debugging purposes we define two initial guard values to protect
 * against accidental POPs and RETURNS.
 *
 * INITIAL_FRAME_OFFSET should allow enough space for all values popped by a
 * RETURN (it will jump to location 0)
 *
 * INITIAL_FRONT_OFFSET is INITIAL_FRONT_OFFSET plus an allowance for a POP
 * to get a zero.
 *
 * These values are expressed as -<no of 32 bit ints> we allow a safety factor
 * of two just to be extra cautious, i.e. if return pops one int we define the
 * INITIAL_FRAME_OFFSET as -2 rather -1.
 */

#define MULTI_TASKING
#define STACK_SIZE MIN_STACK_SIZE*4

#define STACK_FRAME 3
#define STACK_FRONT 4

#define INITIAL_FRAME_OFFSET (-4)
#define INITIAL_FRONT_OFFSET (-6)
typedef int ansa_RegDump[6];

extern int stack_dump (/* ansa_Reg_Dump env */);
extern void stack_switch (/* ansa_RegDump env, int val */);

#endif  /* __STACK_INCLUDED__ */
```

- **include/machine**
  Plik convex.h jest zgodny z plikiem sparc.h za wyjątkiem definicji dwóch typów: **ansa_LongCardinal** i **ansa_LongInteger**. Oba typy wykorzystują posiadany przez Convex typ **long long int** określający 64-bitowe liczby całkowite.
  Convex 3200 jest maszyną typu Big-Endian.

  ansa/gen_dist/include/machine/convex.h

```
/* Header Start */
/*
 *  Copyright 1989 APM Limited (ISA Project)
 *
 *  File: sparc.h
 *  Author:
 *  Date Created:
 *
 *  General Description:
 *    Machine specific values and constants for sparc.
 *
 *  $Header: /disc2/usr/local/src/ANSAware4.0/master/include/machine/RCS/
sparc.h,v 1.1 1991/07/10 09:15:05 can Exp $
 *
 */
/* Header End */

#ifndef __MACHINE_INCLUDED__
#define __MACHINE_INCLUDED__

/* Various common types */

#ifndef _BASIC_TYPES_
#define _BASIC_TYPES_
typedef int ansa_Boolean;
typedef unsigned short int  ansa_ShortCardinal; /* must be 16 bit */
typedef unsigned long int  ansa_Cardinal; /* must be 32 bit */
typedef unsigned long long int  ansa_LongCardinal; /* should be 64 bit */
typedef short int ansa_ShortInteger; /* 16 bit */
typedef long int ansa_Integer; /* 32 bit */
typedef long long int ansa_LongInteger; /* should be 64 bit */
typedef float ansa_Real;
typedef double ansa_LongReal;
typedef char *ansa_String;
typedef unsigned char ansa_Octet;
typedef char ansa_Char;
#endif /* _BASIC_TYPES_ */

/*
 * define LITTLE_ENDIAN if that is the case for this system
 * VAX, 80x86, ARM
 */
/*#define LITTLE_ENDIAN*/

#endif /* __MACHINE_INCLUDED__ */
```

- **include/stub/MoveMacs**
  Zarówno plik convexIEEE.h jak i m_convexIEEE.h są identyczne z plikami sparcIEEE.h i m_sparcIEEE.h.

  ansa/gen_dist/include/stub/MoveMacs/convexIEEE.h

```
/* Header Start */
/*
 *  Copyright 1990 APM Limited (ISA Project)
 *
 *  File: sparcIEEE.h
 *  Author:
 *  Date Created:
 *
 *  General Description:
 *    Marshall/Unmarshall macros for sparc systems with IEEE fp.
 *
 *  $Header: /disc2/usr/local/src/ANSAware4.0/master/include/stub/MoveMacs/RCS/
sparcIEEE.h,v 1.1 1990/04/16 13:50:09 can Exp $
 *
 */
/* Header End */


#ifndef __MOVEMACS_INCLUDED__
#define __MOVEMACS_INCLUDED__

/* This file is #included within the generated stub. */

#define Move0808(a,b) (*(char *)(b) = *(char *)(a))

#define Move1616(a,b) (*(char *)(b) = *(char *)(a), \
*((char *)(b)+1) = *((char *)(a)+1))

#define Move3232(a,b) (*(char *)(b) = *(char *)(a), \
   *((char *)(b)+1) = *((char *)(a)+1), \
   *((char *)(b)+2) = *((char *)(a)+2), \
   *((char *)(b)+3) = *((char *)(a)+3))

#define Move6464(a,b) (*(char *)(b) = *(char *)(a), \
   *((char *)(b)+1) = *((char *)(a)+1), \
   *((char *)(b)+2) = *((char *)(a)+2), \
   *((char *)(b)+3) = *((char *)(a)+3), \
*((char *)(b)+4) = *((char *)(a)+4), \
   *((char *)(b)+5) = *((char *)(a)+5), \
   *((char *)(b)+6) = *((char *)(a)+6), \
   *((char *)(b)+7) = *((char *)(a)+7))

#define MarshallEnum(a,b,p,s) {if(s<4) goto toobig;Move3232(a,b);p+=4;s-=4;}
#define MarshallBool(a,b,p,s) {if(s<4) goto toobig;Move3232(a,b);p+=4;s-=4;}
#define MarshallSCard(a,b,p,s) {if(s<2) goto toobig;Move1616(a,b);p+=2;s-=2;}
#define MarshallCard(a,b,p,s) {if(s<4) goto toobig;Move3232(a,b);p+=4;s-=4;}
#define MarshallLCard(a,b,p,s) {if(s<4) goto toobig;Move3232(a,b);p+=4;s-=4;}
#define MarshallSInt(a,b,p,s) {if(s<2) goto toobig;Move1616(a,b);p+=2;s-=2;}
#define MarshallInt(a,b,p,s) {if(s<4) goto toobig;Move3232(a,b);p+=4;s-=4;}
#define MarshallLInt(a,b,p,s) {if(s<4) goto toobig;Move3232(a,b);p+=4;s-=4;}
#define MarshallReal(a,b,p,s) {if(s<4) goto toobig;Move3232(a,b);p+=4;s-=4;}
```

```
#define MarshallLReal(a,b,p,s) {if(s<8) goto toobig;Move6464(a,b);p+=8;s-=8;}
#define MarshallOctet(a,b,p,s) {if(s<1) goto toobig;Move0808(a,b);p+=1;s-=1;}
#define MarshallChar(a,b,p,s) {if(s<1) goto toobig;Move0808(a,b);p+=1;s-=1;}

#define UnmarshallEnum(a,b) (Move3232(a,b),4)
#define UnmarshallBool(a,b) (Move3232(a,b),4)
#define UnmarshallSCard(a,b) (Move1616(a,b),2)
#define UnmarshallCard(a,b) (Move3232(a,b),4)
#define UnmarshallLCard(a,b) (Move3232(a,b),4)
#define UnmarshallSInt(a,b) (Move1616(a,b),2)
#define UnmarshallInt(a,b) (Move3232(a,b),4)
#define UnmarshallLInt(a,b) (Move3232(a,b),4)
#define UnmarshallReal(a,b) (Move3232(a,b),4)
#define UnmarshallLReal(a,b) (Move6464(a,b),8)
#define UnmarshallOctet(a,b) (Move0808(a,b),1)
#define UnmarshallChar(a,b) (Move0808(a,b),1)

#endif /* __MOVEMACS_INCLUDED__ */
```

ansa/gen_dist/include/stub/MoveMacs/m_convexIEEE.h

```
/* Header Start */
/*
 *  Copyright 1989-1992 APM Limited (ISA Project)
 *  Copyright BNR Europe Ltd
 *
 *  File:
 *  Author:
 *  Date Created:
 *
 *  General Description:
 * -[un]marshall routines for the builtin types.
 * -they are encoded in terms of the existing "MoveMacs.h",
 * to hopefully ensure conformant behaviour.
 *
 *  $Header: /disc2/usr/local/src/ANSAware4.0/master/include/stub/MoveMacs/RCS/
m_sparcIEEE.h,v 1.4 1992/04/07 08:51:14 can Exp $
 *
 */
/* Header End */

#ifndef _m_builtin_h
#define _m_builtin_h

#include "MoveMacs.h"
#include "AWmacros.h"

/* It is here assumed that "ansa_prefix" is "ansa_". */

#define marshall__N(buf, p, N, MOVER)\
     (buffer_require((buf),N) ||(MOVER(p, (buf)->ptr), (buf)->used += N, \
```

```
     (buf)->unused -= N, (buf)->ptr += N, 0))

#define marshall_enum(buf, des)         marshall__N(buf, (des), 4, Move3232)
#define marshall_ansa_Boolean(buf, b) marshall__N(buf, (b), 4, Move3232)

#define marshall_ansa_Cardinal(buf, n)  marshall__N(buf, (n), 4, Move3232)
#define marshall_ansa_ShortCardinal(buf, n) marshall__N(buf, (n), 2, Move1616)
#define marshall_ansa_LongCardinal(buf, n) marshall__N(buf, (n), 4, Move3232)

#define marshall_ansa_Integer(buf, n) marshall__N(buf, (n), 4, Move3232)
#define marshall_ansa_ShortInteger(buf, n) marshall__N(buf, (n), 2, Move1616)
#define marshall_ansa_LongInteger(buf, n) marshall__N(buf, (n), 4, Move3232)

#define marshall_ansa_Real(buf, n)         marshall__N(buf, (n), 4, Move3232)
#define marshall_ansa_LongReal(buf, n)          marshall__N(buf, (n), 8, Move6464)

#define marshall_ansa_Octet(buf,b)         marshall__N(buf, (b), 1, Move0808)
#define marshall_ansa_Char         marshall_ansa_Octet

#define unmarshall__N(buf, p, N, MOVER)\
     ((buffer_readable(buf) >= N) ?  (MOVER((buf)->ptr, (p)), (buf)->ptr += N, 0) : 1)

/* unrobust, but slighty faster is..
   #define unmarshall__N(buf, p, N, MOVER)\
         (MOVER((buf)->ptr, (p)), (buf)->ptr += N, 0)
*/

#define unmarshall_enum(buf, des) unmarshall__N(buf, (des), 4, Move3232)
#define unmarshall_ansa_Boolean(buf, b) unmarshall__N(buf, (b), 4, Move3232)

#define unmarshall_ansa_Cardinal(buf, n)  unmarshall__N(buf, (n), 4, Move3232)
#define unmarshall_ansa_ShortCardinal(buf, n)
unmarshall__N(buf, (n), 2, Move1616)
#define unmarshall_ansa_LongCardinal(buf, n) unmarshall__N(buf, (n), 4, Move3232)

#define unmarshall_ansa_Integer(buf, n) unmarshall__N(buf, (n), 4, Move3232)
#define unmarshall_ansa_ShortInteger(buf, n) unmarshall__N(buf, (n), 2, Move1616)
#define unmarshall_ansa_LongInteger(buf, n) unmarshall__N(buf, (n), 4, Move3232)

#define unmarshall_ansa_Real(buf, n)         unmarshall__N(buf, (n), 4, Move3232)
#define unmarshall_ansa_LongReal(buf, n) unmarshall__N(buf, (n), 8, Move6464)

#define unmarshall_ansa_Octet(buf,b)         unmarshall__N(buf, (b), 1, Move0808)
#define unmarshall_ansa_Char         unmarshall_ansa_Octet

#endif /* _m_builtin_h */
```

- **include/config**
  Nie wymagał żadnych modyfikacji

- **include/opsys**
  Plik odpowiada prawie całkowicie plikowi bsd.h. Dodano jedynie definicję sygnału

SIGCLD, który w założeniu powinien odpowiadać sygnałowi SIGHCLD na komputerze Sun.

ansa/gen_dist/include/opsys/convex.h

```
#define        SIGCLD              20
```

- **src/ansa/capsule/stack.cpp**
  Poniższy fragment zawiera definicję procedur **stack_dump** i **stack_switch** służących do przełączania stosu. Zostały one dopisane do pliku stack.cpp z powodu niemożności skorzystania z sytemowych procedur **setjmp** i **longjmp**.
  Procedura **stack_dump** zapamiętuje, a **stack_switch** odtwarza następujące rejestry procesora:

SP	- Stack Pointer
PSW	- Processor Status Word
FP	- Frame Pointer
PC	- Program Counter

ansa/gen_dist/src/ansa/capsule/stack.cpp

```
#if defined(__convex__)
        .globl  _stack_dump
        .globl  _stack_switch
        .fpmode ieee
        .text

_stack_dump:
        ld.w    0(ap),a1
        ld.w    8(fp),s0
        st.w    s0,12(a1)
        ld.w    4(fp),s0
        st.w    s0,8(a1)
        ld.w    0(fp),s0
        st.w    s0,4(a1)
        mov     sp,s0
        add.w   #16,s0
        st.w    s0,16(a1)
        sub.l   s0,s0
        rtn
```

```
_stack_switch:
        ld.w    0(ap),a1
        ld.w    12(a1),fp
        ld.w    16(a1),sp
        ld.w    8(a1),a3
        mov     a3,psw
        ld.w    4(a1),a2
        jmp     0(a2)
#endif
```

- **include/capsule/machcpp.h**
  Plik ten nie został wymieniony jako wymagający modyfikacji w rozdziale 11 podręcznika **ANSA System Manager's Guide**. Jednak dla przeniesienia oprogramowania konieczne jest uzupełnienie go o definicję komputera Convex 3200.

  ansa/gen_dist/include/capsule/machcpp.h

```
/* Header Start */
/*
 *  Copyright 1989,1990,1991 APM Limited (ISA Project)
 *
 *  File: machcpp.h
 *  Author:
 *  Date Created: 6/5/91
 *
 *  General Description:
 *    CPP definitions for all of the architectures supported by ANSAware.
 *    These definitions are used as values for the macro
 *    _ansa_arch_ supplied on every compiler command line.
 *    Care is taken to avoid clashes between these values
 *    and other often used macros (i.e. we use _aw_vax and not vax!).
 *    _ansa_arch_ is set to the value specified by the Architecture
 *   imake macro defined in the macros file for a given platform.
 *   For each value of this macro there must be corresponding definition
 *   in this file.
 *
 *  $Header: /disc2/usr/local/src/ANSAware4.0/master/include/capsule/RCS/
machcpp.h,v 1.2 1991/09/24 17:11:23 can Exp $
 *
 */
/* Header End */

#ifndef __MACHCPP_INCLUDED__
#define __MACHCPP_INCLUDED__

#define _aw_vax  1   /* DEC VAX (ultrix and vms) */
#define _aw_arm  2   /* Acorn ARM */
```

```
#define _aw_hp300 3   /* HP 300 series */
#define _aw_hp400 4   /* HP 400 series */
#define _aw_hp700 5   /* HP 700 series */
#define _aw_hp800 6   /* HP 800 series */
#define _aw_w68k  7   /* Wanda 68k */
#define _aw_wvax  8   /* Wanda vax */
#define _aw_decmips  9   /* DEC mips */
#define _aw_sun3      10  /* sun 3 series (68k based) */
#define _aw_sun4 11  /* sun 4 series (sparc based) */
#define _aw_convex3200  12  /* convex 3200 series */

#endif /* __MACHCPP_INCLUDED__ */
```

Rozdział 2

2.1 Przygotowanie środowiska do kompilacji systemu ANSA

W niniejszej części opisano problemy związane z dołożeniem nowej platformy sprzętowej jak i problemy związane z wygenerowniem środowiska do kompilacji systemu ANSA. Poprawki zostały przedstawione jako wynik działania programu diff na wersji oryginalnej i zmodyfikowanej.

- **master/util/imake/imake.c**

```
136c136
< #if defined(sun) || defined(hpux) || defined(macII) || defined(CRAY)
---
> #if defined(sun) || defined(hpux) || defined(macII) || defined(CRAY) || \
      defined(__convex__)
```

Poprawka związana z definicją identyfikatora REDUCED_TO_ASCII_SPACE

```
172a173,174
>  "-pcc",
>  "-Uunix",
```

Poprawki związane z preprocesorem C **/lib/cpp**.
Opcja "-pcc" powoduje zachowanie zgodności z preprocesorami C, które nie stosowały się do standardu ANSI C. W szczególności poprawka powoduje, że komentarze nie będą zastępowane spacją.
Opcja "-Uunix" usuwa defincję symbolu "unix" wprowadzoną przez opcję "-pcc". Zdefiniowanie symbolu "unix" powoduje błędy w czasie generowania plików *makefile* dla systemu ANSA.

```
195a198,201
> #endif
>
> #ifdef convex3200_cos_10_0
>  "-Dconvex3200_cos_10_0"
```

Definicja unikalnego symbolu dla preprocesora C określającego nazwę platformy dla komputera Convex.

- **master/util/config/Imake.tmpl**

```
100a101,107
> #ifdef convex3200_cos_10_0
> #define MacroIncludeFile "convex3200_cos_10.0.macros"
> #define MacroFile convex3200_cos_10.0.macros
> #define ConvexArchitecture
> #undef convex3200_cos_10_0
> #endif /* convex */
>
```

Definicje symboli dla nowej platformy oraz nazwy tzw. pliku "macros", przedstawiono poniżej.

- **master/util/config/convex3200_cos_10.0.macros**
  Plik zawiera definicję symboli niezbędnych do prawidłowego wygenerowania środowiska do kompilacji systemu ANSA.

```
/* Header Start */
/*
 *  Copyright 1989 APM Limited (ISA Project)
 *
 *  File: sun4_sos_4.1.macros
 *  Author:
 *  Date Created: 03/12/89
 *
 *  General Description:
 *    Definitions for Sun4 SunOS4.0
 *
 *  $Header: /disc2/usr/local/src/ANSAware4.0/master/util/config/RCS/
sun4_sos_4.1.macros,v 1.1 1992/01/08 11:12:41 can Exp $
 *
 */
/* Header End */

/**/# platform version: $Header: /disc2/usr/local/src/ANSAware4.0/master/util/
config/RCS/sun4_sos_4.1.macros,v 1.1 1992/01/08 11:12:41 can Exp $

/*
 * SET VERSION NUMBERS BEFORE MAKING MAKEFILES
 *
 * If you change either of the following numbers, you will need to do a
 * full build (make World) at the top of the sources....
 */
#define ConvexPlatform    YES
```

```
#define OSName            ConvexOS 10.0
#define OSMajorVersion    10
#define OSMinorVersion    0


/* Header Start */
/*
 *  Copyright 1989 APM Limited (ISA Project)
 *
 *  File: sunos.macros
 *  Author:
 *  Date Created: 03/12/89
 *
 *  General Description:
 *    Common definitions for all versions of SunOS.
 *
 *  $Header: /disc2/usr/local/src/ANSAware4.0/master/util/config/RCS/
sunos.macros,v 1.7 1991/09/26 14:33:39 can Exp $
 *
 */
/* Header End */

/**/# platform: $Header: /disc2/usr/local/src/ANSAware4.0/master/util/config/RCS/
sunos.macros,v 1.7 1991/09/26 14:33:39 can Exp $
/**/# operating system: OSName

 BOOTSTRAPCFLAGS =
             AS = as
             CC = cc -fi -U__stdc__
            CPP = /lib/cpp
           LINT = lint
        INSTALL = install -c
             LD = $(CC)
           YACC = yacc
            LEX = lex
            AWK = awk
            CMP = cmp
           TAGS = ctags
             RM = rm -f
             MV = mv
             CP = cp
          TOUCH = touch
             LN = ln -s
         RANLIB = ranlib
 RANLIBINSTFLAGS = -t
             AR = ar rv
             LS = ls
       LINTOPTS = -axz
    LINTLIBFLAG = -C
           MAKE = make
   STD_IDLFLAGS = $(DEF_IDLFLAGS)
   STD_DPLFLAGS = $(DEF_DPLFLAGS)
 STD_CPP_DEFINES =
```

```
     STD_DEFINES = $(DEF_DEFINES)
    STD_INCLUDES = $(DEF_INCLUDES)
STD_IDL_INCLUDES = $(DEF_IDL_INCLUDES)

/* define this as you like for normal library compilation */
#define LibraryDefines $(DEF_DEFINES)

#define OperatingSystem _ansa_unix_
#define OperatingSystemCore unix
#define UnixCore YES
#define SysFirstLibraries
#define SysLastLibraries
#define NetworkLibraries
#define DefaultOptFlag      -O

#define OSPlatformFlags -D_ansa_convexos_

#define Architecture _aw_convex3200
```

Oprócz oczywistych poprawek związanych z nową platformą, szczególną uwagę należy zwrócić na definicję następujących symboli:

CC = cc -fi -U__stdc__

Opcja “-fi” kompilatora C powoduje wykonywanie operacji zmiennoprzecinkowych w standardzie IEEE.
Opcja “-U__stdc__” powoduje usunięcie definicji symbolu “__stdc__”. Symbol “__stdc__” nie oznacza, że kompilator stosuje się do standardu ANSI ale że kompiluje w stylu ANSI.

INSTALL = install -c

Opcja “-c” powoduje że binaria sa kopiowane (**cp**), a nie przesuwane (**mv**).

Rozdział 3

3.1 Kompilacja systemu ANSA

Poniżej opisane zostały problemy związane z kompilacją systemu ANSA na komputerze Convex. Problemy te wynikają z różnic pomiędzy kompilatorami C na maszynach typu Sun i Convex. Poprawki do plików źródłowych zostały zaprezentowane jako wynik działania programu **diff** na wersji oryginalnej i źródłowej.

- **Podstawowe parametry kompilacji**
  Poniższy plik określa podstawowe parametry z którymi była budowana ANSA. Przykładowe pliki dla różnych architektur znajdują się w katalogu **examples**

```
# Header Start
#
#  Copyright 1990 APM Limited (ISA Project)
#
#  File: apm_sun4_sos_4.1
#  Author:
#  Date Created: 25/04/90
#
#  General Description:
#    build parameters used at apm for sun4's
#
#  $Header: /disc2/usr/local/src/ANSAware4.0Dist/examples/RCS/
apm_sun4_sos_4.1,v 1.4 1992/05/05 15:05:28 can Exp $
#
# Header End
# name of platform
convex3200_cos_10.0
# use default shadow tree

# pathname of top level directory
/usr/tmp
# read only installation point
/mnt/yppawlus/ansa/install/convex3200_cos_10.0
# read only installation point suffix (i.e. none)

# read/write installation point
```

```
/mnt/yppawlus/ansa/tmp/awdev
# read/write installation point suffix (i.e. none)

# default MPS's (IPC and UDP)
UDP TCP
# directory for fifos
# if you are not using IPC then you must not
# supply this answer, therefore delete the line
# following these comments if your not using IPC
# master trader name
krak
# master ip address
149.156.1.1
# local trader name
krak
# no need for ip address, it's the same as the masters
# if you use different master and local traders you must
# give an ip address for the local trader after this comment and
# before the next prompt.
# use default CC
cc -fi -U__stdc__
# use default CC options

# build capsulep YES/NO
YES
# confirm build parameters

# do not install
no
```

Powody użycia dodatkowych opcji kompilatora C: "cc -fi -U__stdc__" zostały opisane w punkcie 2.1 .

- **master/include/ansi/AWstdio.h**
  Rozważana grupa poprawek jest związana z użyciem zmodyfikowanych wersji plików typu **include** systemu operacyjnego Convex OS. Konieczność modyfikacji plików **include** wynikała z faktu, że pliki te zawieraja dyrektywę **#error**, która nie jest rozumiana przez program **imake**.
  Alternatywnym rozwiązaniem byłoby rozszerzenie zakresu rozpoznawanych przez program **imake** dyrektyw preprocesora C o dyrektywę **#error**.

```
27a28,30
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "stdio.h"
> #else
28a32
> #endif
31a36,38
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "stdio.h"
```

```
> #else
32a40,41
> #endif
>
```

```
60c69,70
< #if defined(_ansa_sunos_) && (_ansa_osmajor_ == 4) && (_ansa_osminor_ == 1)
---
> #if defined(_ansa_sunos_) && (_ansa_osmajor_ == 4) && (_ansa_osminor_ == 1) \
>  || defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
```

Przyjęcie zgodnej z systemem Sun OS definicji funkcji sprintf.

- **convex3200_cos_10.0/include/stdio.h**

```
7,11d6
< #if __stdc__ /* check for unsupported extension */
< #  if defined(_CONVEX_SOURCE) && !defined(_POSIX_SOURCE)
< #    error _CONVEX_SOURCE may not be defined without _POSIX_SOURCE
< #  endif
< #endif
```

```
222a218,219
> #   define _IOFBF 01000  /* dodano dla kompilacji kb.c, alternatywa
>                                  zmiana _IOFBF w kb.c na _IONBF */
```

Patrz uwaga w komentarzu powyżej. Zdefiniowanie stałej _IOFBF powoduje pełne buforowanie operacji wejścia/wyjścia.

- **master/include/ansi/AWmath.h**

```
21a22,24
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "math.h"
> #else
22a26
> #endif
44a49,51
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "math.h"
> #else
45a53
> #endif
```

- **convex3200_cos_10.0/include/math.h**

```
4,8d3
< #if __stdc__ /* check for unsupported extension */
< #  if defined(_CONVEX_SOURCE) && !defined(_POSIX_SOURCE)
< #    error _CONVEX_SOURCE may not be defined without _POSIX_SOURCE
< #  endif
< #endif
128c123
< #
---
>
293c288
< #
---
>
```

- **master/include/ansi/AWstring.h**

```
21a22,24
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "string.h"
> #else
22a26,27
> #endif
>
```

- **convex3200_cos_10.0/include/string.h**

```
6,10d5
< #if __stdc__ /* check for unsupported extension */
< #  if defined(_CONVEX_SOURCE) && !defined(_POSIX_SOURCE)
< #    error _CONVEX_SOURCE may not be defined without _POSIX_SOURCE
< #  endif
< #endif
```

- **master/include/ansi/AWtime.h**

```
49a50,52
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "time.h"
> #else
50a54
> #endif
```

- **convex3200_cos_10.0/include/time.h**

```
7,11d6
< #if __stdc__ /* check for unsupported extension */
< #  if defined(_CONVEX_SOURCE) && !defined(_POSIX_SOURCE)
< #    error _CONVEX_SOURCE may not be defined without _POSIX_SOURCE
< #  endif
< #endif
```

- **master/util/makedepend/def.h**

```
35a36,38
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "ctype.h"
> #else
36a40
> #endif
```

- **master/include/capsule/mifref.hd**

```
53a54,56
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "ctype.h"
> #else
54a58,59
> #endif
>
```

- **convex3200_cos_10.0/include/ctype.h**

```
7,11d6
< #if __stdc__ /* check for unsupported extension */
< #  if defined(_CONVEX_SOURCE) && !defined(_POSIX_SOURCE)
< #    error _CONVEX_SOURCE may not be defined without _POSIX_SOURCE
< #  endif
< #endif
```

- **master/include/ansi/AWunistd.h**

```
51c51,52
< #if ! (defined(_ansa_sunos_) && (_ansa_osmajor_ == 4) && (_ansa_osminor_ == 1))
---
> #if ! (defined(_ansa_sunos_) && (_ansa_osmajor_ == 4) && (_ansa_osminor_ == 1) \
>  || defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor == 0))
```

Defincja typów **pid_t** i **mode_t**.

```
86c87
< #ifdef _ansa_hpux_
---
> #ifdef _ansa_hpux_
88a90,92
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> extern pid_t fork P(( void ));
> #else
90a95
> #endif

93a99
> #if ! (defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0))
94a101
> #endif
```

System posiada definicję typu **ssize_t**.

- **master/include/ansi/AWsocket.h**

```
24c24
< #if defined(_ansa_riscix_) || defined(_ansa_ultrix_) || defined(_ansa_hpux_8_0_)
---
> #if defined(_ansa_riscix_) || defined(_ansa_ultrix_) || defined(_ansa_hpux_8_0_) || \
defined(_ansa_convexos_)
```

Poprawka związana z definicją identyfikatora _INET_INADDR_AS_LONG_, który określa sposób przechowywania adresu IP.

- **master/util/stubc/main.c**

```
44c44
< "/*\n * Generated by ‘stubc $Revision: 1.15 $’\n *\tfrom ‘%s’\n *\ton ‘%s";
---
> "/*\n * Generated by ‘stubc $Revision: 1.15 $’\n *\tfrom ‘%s’\n *\ton ‘%s";/**/
```

Poprawka związana z błędem kompilatora C na komputerze Convex. Kompilator nie rozpoznaje symbolu zamknięcia komentarza wewnątrz łańcucha znaków.

- **master/util/makedepend/main.c**

```
110c110,112
<
---
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #define SIGNALRETURNSINT
> #endif
```

Symbol SIGNALRETURNSINT umożliwia zdefiniowanie typu funkcji obsługującej sygnały jako **int**.

- **master/src/ansa/capsule/CORE_unix/capsule.c**

```
42a43,45
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "signal.h"
> #else
43a47
> #endif
45a50,52
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "signal.h"
> #else
46a54
> #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

- **convex3200_cos_10.0/include/signal.h**

```
7,11d6
< #if __stdc__ /* check for unsupported extension */
< #  if defined(_CONVEX_SOURCE) && !defined(_POSIX_SOURCE)
< #    error _CONVEX_SOURCE may not be defined without _POSIX_SOURCE
< #  endif
< #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

- **master/src/ansa/capsule/capntfy.dpl**

```
27a28,30
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "signal.h"
> #else
```

```
28a32
> #endif
140a145,147
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
>     int     (*intfcn)(), (*trmfcn)();
> #else
141a149
> #endif
```

Zadeklarowanie, że w/w funkcje są typu **int**, a nie **void**.

- **master/src/ansa/test/callsrv.c**

```
26a27,29
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "signal.h"
> #else
27a31
> #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

- **master/src/ansa/trader/trader.dpl**

```
31a32,34
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "signal.h"
> #else
32a36
> #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

- **master/src/ansa/factory/server.dpl**

```
21a22,24
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "signal.h"
> #else
22a26
> #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

- **master/src/ansa/notify/GrReaper.dpl**

```
24a25,27
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "ctype.h"
> #else
25a29
> #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

```
50a55,59
> #ifdef _ansa_convexos_
> #include <sys/types.h>
> #include <dirent.h>
> #endif
>
```

Włączenie plików systemowych **types.h** i **dirent.h**.

```
108a118,120
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "signal.h"
> #else
109a122
> #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

```
137a151,153
> #ifdef _ansa_convexos_
>  struct dirent *dp;
> #endif
```

```
200c216
<  for (dp = readdir(dirp); dp != NULL; dp = readdir(dirp))
---
>  for (dp = readdir(dirp); dp != NULL ; dp = readdir(dirp))
```

Definicja typu zmiennej **dp** właściwa dla systemu Convex OS.

- **master/src/ansa/capsule/buffer.c**

```
22c22,24
<
---
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "assert.h"
> #else
23a26
> #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

- **convex3200_cos_10.0/include/assert.h**

```
4,9d3
< #if __stdc__ /* check for unsupported extension */
< #  if defined(_CONVEX_SOURCE) && !defined(_POSIX_SOURCE)
< #    error _CONVEX_SOURCE may not be defined without _POSIX_SOURCE
< #  endif
< #endif
<
```

- **master/src/ansa/nodemgr/client.dpl**

```
24a25,28
>
> #if defined(_ansa_convexos_) && (_ansa_osmajor_ == 10) && (_ansa_osminor_ == 0)
> #include "assert.h"
> #else
25a30
> #endif
```

Poprawka związana z dyrektywą **#error**, patrz wyżej.

- **master/src/ansa/capsule/Imakefile**

```
32c32
<        CPPFLAGS = -P $(OSPLATFORMFLAGS)
---
>        CPPFLAGS = -P -pcc $(OSPLATFORMFLAGS)
```

Opcja "-pcc" powoduje zachowanie zgodności z preprocesorami C, które nie stosowały się do standardu ANSI C.

3.2 Sprawdzenie poprawności instalacji systemu ANSA

Dla oprogramowania ANSA wygenerowanego w sposób opisany w niniejszym opracowaniu wykonano serię testów zaproponowannych w rozdziale 7 podręcznika ANSA System Manager's Guide. Przeprowadzone testy wykazały poprawne działanie wygenerowanego oprogramowania.