



Title	大阪大学大型計算機センターのシステム
Author(s)	システム管理掛
Citation	大阪大学大型計算機センターニュース. 1997, 104, p. 2-40
Version Type	VoR
URL	https://hdl.handle.net/11094/66207
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

大阪大学大型計算機センターのシステム

システム管理掛

まえがき

大阪大学大型計算機センターのシステム構成をはじめ、計算機の一般的な利用形態、スーパーコンピュータ SX-4/64M2(以下、SX-4)で利用できる言語、ライブラリ、プログラムのチューニングのための開発支援ツール、アプリケーションソフトウェア等の説明を行い、最後に UNIX サブシステム Onyx 及びワークステーションで利用できるソフトウェアを紹介します。

1. システム構成 2. 利用方法 3. 言語 4. 数値ライブラリ
5. プログラミング支援環境 6. プログラム開発支援 7. デバック支援
8. エディタ 9. グラフィック 10. アプリケーション 11. ワークステーション
12. 電子マニュアル 13. 電子メールと電子ニュース 14. インターネットサービス
15. FAX サービス 16. マルチメディア情報蓄積システム

1. システム構成

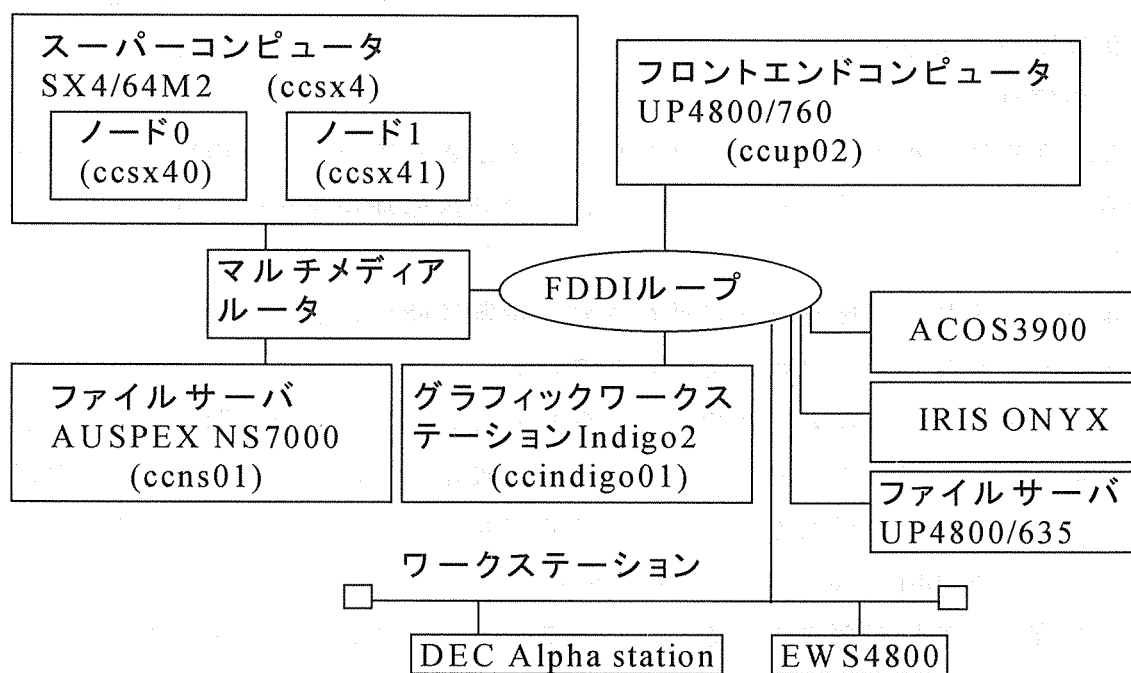


図 計算機システム接続概略

1.1. スーパーコンピュータ SX-4/64M2

1 ノード 32 台の CPU と 8GB の主記憶、16GB の拡張記憶を有する共有メモリ方式のベクトル並列型スーパーコンピュータで 2 ノード構成となっています。オペレーティングシステムには UNIX System V に準拠した SUPER-UX が搭載されています。

1.2. フロントエンドコンピュータ UP4800/760

MIPS 社の CPU である R10000 が 4 台搭載されたワークステーションで、オペレーティングシステムには UNIX System V に準拠した UX/4800 が搭載されています。スーパーコンピュータ SX-4 のクロス開発環境として、FORTRAN90/SX クロスコンパイラ、C/SX クロスコンパイラが利用できます。IMSL ライブラリも使用できます。1/2 インチカートリッジ磁気テープ装置、1/4 インチカートリッジ磁気テープ装置等の周辺機器が接続されています。

1.3. グラフィックワークステーション Indigo2

MIPS 社の CPU である R10000 が搭載されたワークステーションで、オペレーティングシステムには UNIX system V に準拠した IRIX 6.2 が搭載されています。AVS 並びに Mathematica、IMSL ライブラリが利用できます。フルカラープリンタ A Color620、フィルムレコーダ、イメージスキャナ、8 ミリ VTR 等の周辺機器が接続されています。また、SX-4 の構造解析、流体解析等のアプリケーションを画像表示（プリ・ポスト処理）する各種ソフトウェアが準備されています。

1.4. ファイルサーバ AUSPEX NS7000

AUSPEX NS7000 は 806GB のディスク容量を持つファイルサーバで、利用者のホームディレクトリが作成されています。

1.5. AlphaStation600 5/333

DEC 社のワークステーションで、Digital UNIX V3.2D のオペレーティングシステムが搭載されています。cpu のクロック速度 333MHz、メモリ容量 512MB、1.44MB フロッピー、DAT、CD-ROM 等の周辺機器が接続されています。使用可能言語は C,C++,Fortran77 などのソフトウェアが利用できます。しかし、センターのメールサーバも兼ねておりますので、メモリの大きなプログラム及び長時間ジョブは ccup02 または SX-4 をご利用ください。

1.6. Onyx

シリコングラフィックス社のワークステーションで、オペレーティングシステムには UNIX system V に準拠した IRIX 5.3 が搭載されています。1GB の主記憶容量、8CPU、176MFLOPS の処理能力、26GB の磁気ディスク容量を有するスーパーグラフィックワークステーションです。グラフィックス処理の周辺装置もイメージスキャナ、フルカラープリンタ、DAT 等、豊富に取り揃えています。

2. 利用方法

ログインとパスワード

SX-4、ワークステーション及びインターネットサービスのログイン名の管理は、これまで 7 桁の登録番号（例えば a69999a）で行なってきました。しかし、支払いコード別に ID を使い分けなければならないという複雑さを解消するため、今年度より、利用者番号（例えば a69999）で管理するよう変更をいたしました。このためパスワードも NIS(Network Information Service)により、一元管理されています。パスワードの変更は yppasswd コマンドで行なってください。なお、利用申請直後は初期パスワードが

設定されていますので速やかに変更してください。

ファイルシステムとホームディレクトリ

ACOS3900 汎用機システムを除き、SX-4 及びワークステーション群のホームディレクトリはファイルサーバ AUSPEX NS7000 上にあり、NFS(Newtwork File System)により共有しています。各利用者のホームディレクトリは実際には ccns01 上の /home/ ccns01/user[1~10]/利用者番号 に存在しています。センターではホームディレクトリを共有するどのマシンからでも /usr1/利用者番号 で参照できるように設定しています。ファイルの容量制限は設けていませんが、1 ファイル最大 2GB までの物理的な制限があります。SX-4 のローカルディスクの容量は 1GB の制限を設けていますが、これを超える場合は SX-4 ファイル拡張申請書が必要です。ローカルディスクのディレクトリ名は /home/local/利用者番号 です。

環境設定ファイル

SX-4 又はワークステーションを初めて利用する場合には初期設定(環境設定)を行う必要があります。まだ、設定されていない方はセンターで準備している環境設定用ファイルをコピーして使いやすいように編集してお使いください。

環境設定用ファイルのコピー

```
% cp -I ~guest/.??* ~/.
```

環境設定ファイル.cshrc、.login を実行

```
% source .cshrc
```

```
% source .login
```

または、センター作成の /usr/local/bin/kankyuu コマンドで初期設定を行なってください。

2.1. SX-4 の利用形態

SX-4 を使用する利用形態として telnet 等で直接接続して会話型で利用する会話型形式とバッチジョブを投入するバッチ処理形式があります。バッチ処理は NQS (Network Queuing System) コマンドを使用します。

2.1.1. SX-4 会話型で利用

会話型で利用できる資源は、メモリ 1GB、CPU 時間 1 時間、CPU 数 1 個となって SX-3 と比べて大幅に増えております。また、接続時間にかかる負担金項目も解除されました。

【会話型の接続例】

```
% telnet ccsx4.center.osaka-u.ac.jp
```

```
login : w69999 (利用者番号)
```

```
passwd : パスワード
```

2.1.2. バッチ処理 (NQS) での利用

NQS のバッチ処理では、ジョブの投入、確認、状況表示、ジョブの削除などの NQS コマンドから構成されています。SX-4 に NQS を利用してバッチジョブを投入する場合、直接 telnet で ccsx4 にログインするか、またはフロントエンドサーバ(ホスト名: ccup02)にログインしてご利用ください。

SX-4 はマルチノード構成になっています。このため、NQS でバッチジョブを投入しますとシステムが負荷の軽い方のノードにジョブを振り分けます。したがって、利用者はジョブがどちらのノードで処理されるか分かりません。このため NQS コマンドの利用方法が SX-3 と少し変わります。具体的には、ジョブの状況表示 (qstatr のコマンド) のオプションが必要になります。

NQS では基本的にバッチジョブをシェルスクリプトで記述します。

【 NQS のオプションの指定方法】

```
#      : コメント
@$-    : オプションの指定
@$     : オプションの終了
```

【シェルスクリプト例】

% cat testnqs	スクリプトの内容
#!/usr/bin/csh(1) コメント
##@\$-q p4 ジョブクラス p4 を指定する
##@\$-lt "0:0:10" CPU 時間 10 秒を上限に指定する
##@\$-o data 標準出力ファイルを指定する
##@\$(2) オプションの終了を示す
cd f77	カレントディレクトリを f77 に移動
f77sx main.f /usr/lib/asl	asl ライブラリを指定する場合に必要な
if(\$status !=0) then	コンパイルエラーがあれば終了
goto DONE	
endif	
setenv F_FILEINF YES	ファイルの入出情報を表示する場合に必要な
setenv F_PROGINF YES	プログラムの実行時の情報を表示する場合に必要な
a.out	実行
DONE:	

(1) ~ (2) : は NQS のオプションを指定している

2.1.2.1. SX-4 での直接利用

(1) qsub コマンド

ジョブを投入するには、qsub コマンドを用います。ジョブはシェルスクリプトで記述し、NQS ではこれをバッチリクエストといいます。詳細については man qsub を参考にしてください。

【形式】 qsub オプション スクリプトファイル

オプションの指定順序は任意で、空白で区切って指定します。

シェルスクリプトの先頭のコメント部に、'@\$'に続けて記述することができます。

【例】 testnqs をバッチジョブとして ccsx4 に投入する

```
% qsub testnqs
% Request 30001.ccsx4 submitted to queue: p4.
```

リクエストが受け付けられたメッセージです。30001.ccsx4 はリクエスト ID で、SX-4 が自動的に発行する連番（リクエスト番号）とホスト名からなります。

【例】 script をジョブクラス p4 に投入する

```
% qsub -q p4 script
```

```
% Request 1.ccsx40 submitted to queue: p4.          リクエスト ID です
```

注意：telnet あるいは rlogin などで ccsx4 にログインしますとホスト名は ccsx40 になります。

[出力結果ファイルの変更]

出力ファイルの指定 (qsub コマンドのオプション -o、-e) がなければ、ジョブを投入したときのカレントディレクトリ配下に出力されます。

【例】 リクエスト名.o リクエスト番号 ----- 標準出力ファイル

 リクエスト名.e リクエスト番号 ----- 標準エラー出力ファイル

(投入時-o,-e オプション使用時)

- ・ オプションで指定したファイルに出力
- ・ #@\$.o data の場合リクエストされたカレント位置の data に出力

[-r オプション]

【例】 リクエスト投入位置にリクエスト名を指定します。

```
% qsub -r prol testjob
```

```
% Request 18043.ccsx4 submitted to queue:p4.
```

```
% prol.o18043    prol.e18043
```

[-mb,-me オプション]

リクエストジョブが実行された開始又は終了時間にメールサーバ ccalpha01 にメールが送信される。

【例】 -mb 又は -me

[-mu オプション]

オプションは user(@文字を含まない)か、user@machine のどちらかの形式で指定します。

【例】 -mu user1 現ホストの user1 にメールを送信するようにする

 -mu user1@host1 ホスト host1 の user1 にメールを送信するようにする

表 qsub の主なオプション (script-file に埋め込みも可)

オプション	機能	例
-q queue-name	キューの指定	-q p8
-lt [[時間:]分:]秒	プロセス毎の CPU 時間の制限	-lt 1:10:30
-lT [[時間:]分:]秒	リクエスト毎の CPU 時間の制限	-lT 1:00:00
-o stdout-filename	標準出力用ファイルを指定	-o output_data
-e stderr-filename	標準エラー出力用ファイルを指定	-e err_data

-mb	実行開始時にメールを送る (ワークステーションからのみ)	-mb
-me	実行終了時にメールを送る (ワークステーションからのみ)	-me
-mu user-name	メールの送信先を指定します	-mu user1
-r request-name	リクエスト名の指定	-r リクエスト名
-jo	JOR を標準出力用ファイルに出力	-jo
-je	JOR を標準エラー出力用ファイルに出力	-je

(2)qstatq コマンド

【例】 実行中のバッチジョブとバッチキューの状態を表示する

% qstatq

NQS (R06.20) BATCH QUEUE SUMMARY HOST: ccsx40										
QUEUE NAME	ENA	STS	PRI/BPR/	TMS /MPR	RLM	TOT	QUE	RUN	WAI	HLD
SUS ARR EXT										
p4	ENA	INA	30/ 80/ 1000/	0	3	0	0	0	0	0
			省略							
p16	ENA	INA	30/ 80/ 1000/	0	3	0	0	0	0	0
<TOTAL>				50	4	0	4	0	0	0

なお、バッチキューのみの情報を表示する場合は、オプション-bをつけます。

% qstatq -b

または、センターで作成した jobq コマンドで両ノードの状況を確認することができます。

% jobq

(3)qstatr コマンド

SX-4 はマルチノードでバッチジョブが処理されますのでオプション -t 2 を指定してください。

サーチレベルを 2 と指定しますとリモートホストに転送されたジョブの状況も表示され、完全な情報が表示されます。リクエスト ID を指定した場合は、このレベル 2 がデフォルトとなります。

【例】 % qstatr -t 2

NQS (R06.20) BATCH REQUEST HOST: ccsx40

REQUEST ID	NAME	OWNER	QUEUE	PRI	NICE	MEMORY	TIME	STT	JID	R
1) ccsx40	testnqs*	w69999	p8	31	0	2212	0.453	RUN	1	-
		省略								

(4)qdel コマンド

投入したジョブをキャンセルします。実行中のジョブをキャンセルするときは、オプション-k を使います。

【例】 % qdel 1.ccsx40

Request 1.ccsx40 is running. 実行中のためキャンセルできませんでした

% qdel -k 1.ccsx40

Request 1.ccsx40 is running, and has been signalled.

2.1.2.2. フロントエンドサーバ(ホスト名 : ccup02)からの利用

【例】telnet で ccup02 にログインし、qsub コマンドでジョブ投入する

```
telnet ccup02
```

```
% qsub -q p4 script1
```

script1 をジョブクラス (p4) で投入する

```
% Request 30001.ccup02 submitted to queue: p4.
```

リクエストが受け付けられたメッセージです。30001.ccup02 はリクエスト ID で、ccup02 が自動的に発行する連番 (リクエスト番号) とホスト名からなります。

パイプキューにより SX-4 にジョブが転送される

[qstatr コマンドは SX-4 の例と同じオプション-t 2 を指定]

```
% qstatr -t 2
```

表示内容はホスト名が ccup02 となり他は SX-4 と同じです。

[qdel コマンド]

```
% qdel -k 1.ccup02
```

または、qsub コマンドをリクエストした同じクライアントであれば、次の様にホスト名を省略できます (ホスト名 ccsx4、ccsx40、ccup02 などの NQS クライアントでは利用可能です。この機能を NQS トラッキング機能と呼びます)。

```
% qdel -k 1
```

2.1.3. NQS で課金先の支払コードを変更する場合

【例】スクリプトファイルに pay コマンドを挿入する

```
% cat script
```

```
#!/usr/bin/csh
```

```
##$-q p4
```

```
##$
```

```
pay b ←課金の支払いコードを b に変更する場合
```

```
f77sx main.f
```

3. 言語

3.1. FORTRAN77/SX の利用方法

FORTRAN77/sx は大規模科学技術計算プログラムに対して、高度な自動ベクトル化、並列化機能や最適化機能を備えています。FORTRAN77/sx は、ANSI FORTRAN 規格 (正式名称は、American National Standard for Information Systems-Programming FORTRAN 3.9-1978) に準拠し、さらに多くの機能を持っています。また、DO ループを対象とした高度な自動ベクトル化機能を備えており、極

めて強力なベクトル化能力を実現しています。

FORTRAN77,90/SX では新たに IEEE 形式の浮動小数点データ形式をサポートし、ワークステーションとの親和性を高めています。更に、浮動小数点データの形式を IEEE 形式で処理することを既定としています。したがって、書式なし入出力の実行においては、各種のデータ形式に変換する場合(FORTRAN の翻訳オプションを変更すること及び浮動小数点データ形式自動変換機能を働かせるための実行時オプション F_UFMTIEEE、F_UFMTFLOAT1、F_UFMTFLOAT2 と要素長調整機能を働かせるオプション F_UFMTADJUST を指定することによって可能です)注意が必要です。

なお、ワークステーションとの親和性を高めるための、f77 コマンドも準備しています。f77 コマンドは f77sx のコンパイラーを用い、f77sx の制御コマンドに変換して利用しています。f90 コマンドは f77 コマンドを継承していますので、今後 fortran90 を活用しようと考えている方は、この f77 をお勧めします。

【コンパイル・リンク】

f77 オプション 入力ファイル名

f77sx オプション 入力ファイル名

FORTRAN のソースプログラムファイルを複数指定する場合は、空白で区切る。

ファイル名のサフィックスは.F 又は.fを指定する。

【例】% **f77 -o prog prog.f**

カレントディレクトリのファイル名 prog.f を翻訳・結合し実行形式ファイル名 prog が作成されます。オプション"o"を省略すると a.out が実行形式ファイル名になります。

【プログラムの実行】

【例】% **prog**

実行に時間がかかる方は、バックグラウンド処理を利用してください。

【例】% **prog &**

【例】% **fg** バックグラウンド処理をフォアグラウンドに戻す場合。

【スクリプトを作成し実行する場合】

【例】% **cat script** script ファイルの内容

#!/usr/bin/csh

a.out << eof

0

2

0

eof

% chmod 755 script スクリプトに実行権を与える

% script &

[1] 2456 バックグラウンドのプロセス id

【実行中のジョブを打ち切りたい場合】

【例】% **kill 2456** バックグラウンドのプロセス id を指定する

【入出力ファイルのリダイレクションを利用する場合】

UNIX のいくつかのコマンドでは、キーボード（標準入力）から入力を受け取るものと仮定しています。また、大部分のコマンドは端末（標準出力）にそれらの出力を表示します。これらの標準入出力をファイルやプログラムに切り替えることをリダイレクションと言っています。リダイレクションの記号を示します

【例】% **a.out < inputdata** 標準入力(装置番号 5)を inputdata に割り当てる

【例】% **a.out > outputdata** 標準出力(装置番号 6)を outputdata に割り当てる

【例】% **a.out << eof** 標準入力に eof を検出するまで入力データとして読む

2

0

eof

【入出力ファイルを指定する場合】

csh の場合

% **setenv F_FFxx data** 装置番号 xx を data に割り当てる

【バッチジョブでの一時入出力ファイルを利用する場合】

csh の場合

% **setenv TMPDIR /jtmp**

% **setenv F_FF08** **data.\$\$**

% **a.out**

テンポラリーファイルの書き込み場所を /jtmp に設定します。環境変数 TMPDIR で示されたパス配下にテンポラリーファイル data.xxxx を生成します。xxxx はシステムが自動生成した番号で、/jtmp 配下に同一の名前が存在しないように指定しています。なお、ジョブクラスによって一時ファイルの容量が制限されています。p4 と p8 では 4GB : p16 では 8GB まで利用できます。

【ライブラリーを利用する場合】

【例】float0(IEEE 形式)の asl ライブラリーを利用して prog.f をコンパイルする

% **f77 prog.f /usr/lib/asl**

【例】float1(標準形式)の imsl ライブラリーを利用して prog.f をコンパイルする

% **f77 prog.f -float1 /usr/lib1/ims1**

なお、ims1 ライブラリーのソースファイルを一覧するにはディレクトリ /usr/local/appli/vni/source/source/ の配下をご覧ください。

表 各種ライブラリ

	浮動小数点データ形式	ライブラリファイル
IMSL ライブラリ	標準形式(float1)	/usr/lib1/imsi
科学技術計算ライブラリ ASL/SX	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib/asl /usr/lib1/asl /usr/lib2/asl
科学技術計算ライブラリ ASLCINT/SX	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib/aslcint /usr/lib1/aslcint /usr/lib2/aslcint
科学技術計算ライブラリ ASLEME/SX	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib/asleme /usr/lib1/asleme /usr/lib2/asleme
数値計算ライブラリ MATHLIB/SX	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib/math /usr/lib1/math /usr/lib2/math
数学ライブラリ BLAS	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib0/libblas.a /usr/lib1/libblas.a /usr/lib2/libblas.a
数学ライブラリ LAPACK	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib0/liblapack.a /usr/lib1/liblapack.a /usr/lib2/liblapack.a
数学ライブラリ LINPACK	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib0/liblinpack.a /usr/lib1/liblinpack.a /usr/lib2/liblinpack.a
数学ライブラリ EISPACK	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib0/libeispack.a /usr/lib1/libeispack.a /usr/lib2/libeispack.a
数学ライブラリ MINPACK	IEEE 形式(float0) 標準形式(float1) 拡張指数形式(float2)	/usr/lib0/libminpack.a /usr/lib1/libminpack.a /usr/lib2/libminpack.a

【 CERNlib を利用する場合】 (float0 のみ)

格納ディレクトリ : /usr/local/appli/CERNlib

libariadne.a	libbvsl.a	libcdlib.a	libcojets.a
libcomis.a	libcspack.a	libdzdoc.a	libepio.a
libeurodec.a	libfatlib.a	libffread.a	libfritiof.a
libgen.a	libgrafX11.a	libhbook.a	libherwig58.a
libhplot.a	libisajet716.a	libjetset74.a	libkapack.a
libkernlib.a	libkuip.a	liblepto63.a	libminuit.a
libpaw.a	libpdf.a	libphotos.a	libphtools.a
libsigma.a	libzebra.a	libzbook.a	

【リンク時の指定方法】

```
f77 -L/usr/local/appli/CERNlib -l~ ...
```

```
f77 /usr/local/appli/CERNlib/lib~.a ...
```

【例】 libariadne.a を利用する場合

```
% f77 -L/usr/local/appli/CERNlib -lariadne test.f
```

```
% f77 /usr/local/appli/CERNlib/libariadne.a test.f
```

【make コマンドを利用する場合】

```
% cat Makefile Makefile の内容
```

```
PROF = -g
```

```
FC = f77
```

```
OBJ = pi2.o sub1.o sub2.o
```

```
test : $(OBJ)
```

```
$(FC) -o test $(OBJ)
```

```
pi2.o : pi2.f
```

```
$(FC) -c pi2.f $(PROF)
```

```
sub1.o : sub1.f
```

```
$(FC) -c sub1.f $(PROF)
```

```
sub2.o : sub2.f
```

```
$(FC) -c sub2.f $(PROF)
```

【例】 % make

関係するファイルの更新状況をみて自動的にコンパイルする、すなわち、実行結果をみて、sub1.fを修正すると、sub1.fだけをコンパイルして、リンクする。

実行は、

```
% test
```

【プログラム実行解析情報を出力する場合】

csh の場合

【例】% setenv F_PROGINF YES

実行時オプション F_PROGINF の値を YES あるいは DETAIL を指定することにより、プログラム実行時間、命令数等のプログラムに関する情報を標準エラー出力に出力することができます。

***** プログラム情報 *****

経過時間(秒)	プログラム実行に要した実実行時間
ユーザ時間(秒)	プログラム実行に要した CPU 時間の内、ユーザーチン実行に要した時間
システム時間(秒)	プログラム実行に要した CPU 時間の内、システムルーチン実行に要した時間
ベクトル命令実行時間(秒)	プログラム実行に要した CPU 時間の内、ベクトル命令実行に要した時間
全命令実行数	プログラム実行時に実行された全命令の個数
ベクトル命令実行数	プログラム実行時に実行されたベクトル命令の個数
ベクトル命令実行要素数	ベクトル命令で実行された演算要素の個数
浮動小数点データ実行要素数	実行された浮動小数点データ演算の個数
MOPS 値	1 秒間に実行された演算数を 100 万単位で示した値
MFLOPS 値	1 秒間に実行された浮動小数点データ実行要素数を 100 万単位で示した値
平均ベクトル長	実行されたベクトル命令の平均のベクトル長
ベクトル演算率(%)	全実行命令の個数の内、ベクトル命令の個数の占める割合
メモリ使用量(MB)	メモリサイズ
最大同時実行可能プロセッサ数	同時に実行可能なプロセッサの数
1 台以上で実行した時間(秒)	プロセッサ n 台以上で実行した時間
2 台以上で実行した時間(秒)	
：	
9 台以上で実行した時間(秒)	
10 台以上で実行した時間(秒)	
イベントビジー回数	イベント待ちが発生した回数
イベント待ち時間(秒)	イベント待ちに要した時間の合計
ロック待ち時間(秒)	ロック待ちが発生した回数
バリアビジー回数	バリア待ちが発生した回数
バリア待ち時間(秒)	バリア待ちに要した時間の合計
MIPS 値	1 秒間に実行された命令数を 100 万単位で示した値
MIPS 値(実行時間換算)	プロセッサ 1 台以上で実行した時間 1 秒当たりに実行された命令数を 100 万単位で示した値
命令キャッシュミス(秒)	命令キャッシュミスにより発生した時間の合計
オペランドキャッシュミス(秒)	オペランドキャッシュミスにより発生した時間の合計
バンクコンフリクト時間(秒)	バンクコンフリクトの時間

3.2. FORTRAN90/SX の利用方法

FORTRAN90/SX は、ISO による国際規格 ISO/IEC 1539:1991 および日本工業規格 JIS FORTRAN (JIS X 3001(通称 Fortran90))に準拠し、さらに多くの拡張機能を持っています。また FORTRAN77/SX と比べた場合の言語仕様上の特長は以下の通りです。

- ①配列同士の演算や配列の動的割付けと解放
- ②構造体を利用するための構造型
- ③動的に割付けられた配列やリスト構造をアクセスするためのポインタ
- ④プログラム単位で共通のデータの宣言や手続きを記述可能にするとともに、仕様の一部を外から隠ぺいすることを可能にするモジュール
- ⑤内部手続きや手続きの再帰呼び出した、FORTRAN90/SX および他社 FORTRAN との互換性についても十分に考慮しています。

また、FORTRAN77/SX コンパイラの最適化、ベクトル化、並列化、自動ベクトル化、自動並列化機能に加えて、FORTRAN90/SX では、基本ブロックを超え、複数のブロック内で命令の並べ換え等の、最適化処理を行います。これにより、スーパースカラ機能をより有効に利用することができます。

利用方法は直接スーパーコンピュータ(ccsx4)にログインし、会話型処理・バッチ処理を行う方法とフロントエンドサーバ(ccup02)システムによるクロスコンパイラの利用方法があります。また、フロントエンドサーバでは単独で f90 が利用できます。

【セルフコンパイラ】(sx4 のコマンド)

【例】% f90 -o test test.f90

ソース test.f90 から実行モジュール test を生成します。

【クロスコンパイラ】(ccup02 のコマンド)

【例】% sxsf90 -o test test.f90

ワークステーション(フロントエンドサーバシステム)でソースファイルの翻訳・結合を行います。コンパイルにはこちらの ccup02 を推奨します。

【注意事項】

- ① f90 オプションの浮動小数点のデータ形式は既定値で float0 になっています。
- ② FORTRAN90/SX コンパイラを用いて作成したオブジェクトファイルを結合する場合には、f90 コマンド(セルフコンパイラ)又は sxsf90(クロスコンパイラ)を使用して下さい。ld コマンド、cc コマンド、f77 コマンドでは、正しく結合できません。また、以下のオプションは、オプションによって結合されるライブラリが異なるため、翻訳を行ったときと同じオプションを指定して結合してください。

f90 オプション

-float0 | -float1 | -float2 (浮動小数点形式の指定)

-P auto | -P multi | -P stack | -P static (並列処理の指定)

-e w | -d w (1ワードのバイト数の指定)

- ③ クロス環境では、sxsf90 コマンド(クロスコンパイラ)から C 言語プリプロセッサ(cpp)、C コンパ

イラ、アセンブラを起動することはできません。

3.3. C 言語の利用方法

言語は、今日では UNIX のシステム記述言語としてだけでなく、各種アプリケーションプログラムや数値計算など、幅広く利用されています。C/SX の言語仕様は、ANSI C(American National Standard for Information Systems-Programming Language C,ANSI/ISO9899:1990[1992])アメリカ国内情報システム標準規格—プログラム言語 C,ANSI/ISO 9899:1990[1992])、ISO による国際規格(International Standard Programming Language-C,ISO/IEC 9899:1990(E))および日本工業規格 プログラム言語 C JISX 3010 に準拠しています。

C/SX コンパイラは、ソースプログラムを解析して、ベクトル命令で実行可能な部分を自動的に検出し、高度なベクトル命令を生成する、自動ベクトル化機能を有しています。この機能により行列演算等を、非常に高速に行うことができます。また、並列化機能を利用することにより、並列化可能な部分を自動的に検出して並列化する自動並列化機能を備えています。さらに、並列化に関する細かい制御を行うための並列化指示行が用意されています。また、cflow,ctrace 等の各種Cプログラム解析コマンドが備えられています。cflow コマンドによるCプログラムフロー・グラフや ctrace コマンドによる実行トレース等の解析情報が得られます。

【形式】% **cc** **option** 入力ファイル名

C ソースプログラムファイルから翻訳・結合を行い、オブジェクトプログラムファイルを作成します。

C ソースファイルを複数指定できます。.c で終わる名前のファイルは、C ソースプログラムと取り扱われ、翻訳が終了しますと、オブジェクトプログラムが生成され、その名前はソースプログラムの.c の代わりに.o がついた名前が残されます。ただし、翻訳の後、結合されると削除されます。-o オプションを指定しなければ、a.out という名前の実行形式のファイルが作成されます。

【例】% **cc** **src.c**

【make コマンドを利用する場合】

【例】% **cat** **Makefile** Makefile の内容

```
DEBUG_FLAG = -h ansi
PROF = -g -p
CC = cc
OBJ = test.o
test : $(OBJ)
      $(CC) -o test $(OBJ) $(PROF)
test.o : test.c yoshi.h
      $(CC) -c $(DEBUG_FLAG) test.c $(PROF)
```

【例】% **make**

関係するファイルの更新状況をみて自動的にコンパイルする

4. 数値ライブラリ

4.1. ASL/SX

数値計算が必要とされるさまざまな分野の数値シミュレーションプログラムの作成を支援する FORTRAN 用科学技術計算ライブラリです。最近の数値解析学の成果を積極的に取り入れ、アルゴリズムは精度および速度で優れたものを採用しており、SX システムのハードウェア性能を十分発揮できるよう、ベクトル／並列処理を最適化しています。

対応する分野として以下のものがあります。

基本行列演算、連立一次方程式、固有値・固有ベクトル、最小二乗法、FFT、スプライン関数、数値積分、数値微分、常微分方程式、積分方程式、近似・補間、特殊関数、方程式の根、極値問題・最適化、乱数

4.2. ASLCINT/SX

ASL/SX の機能を C 言語プログラムから利用するための C 言語インタフェースライブラリです。

4.3. ASLEME/SX

利用可能な主記憶容量を超えるような大規模データを外部記憶装置に置き、大規模な問題を計算するためのサブルーチンライブラリであり、次の分野に対応します。

基本行列演算、固有値・固有ベクトル、FFT

4.4. IMSL ライブラリ

IMSL ライブラリは、900 個のプログラムからなる数値計算、統計、特殊関数の FORTRAN サブルーチンライブラリで、世界で 12,000 以上の企業・大学で使用されています。行列、微分方程式、非線形代数、スプライン補間や基本統計、回帰分析はもちろん、カルマンフィルタ、ARAMA モデル、ノンパラメトリック計算など、他にも多くの機能を提供します。

— 英語オンラインマニュアル (WWW 上) 及びマニュアルを参照してください —

4.5. NAG FORTRAN LIBRARY

NAG FORTRAN LIBRARY は、科学技術計算用のサブルーチン群です。統計、時系列解析、行列演算、関数の補間と当てはめ数値微積分など、科学技術計算分野の大部分を 900 に及ぶサブルーチンがサポートします。

4.6. LINPACK

LINPACK はいろいろな種類の連立 1 次方程式を解く FORTRAN サブルーチン群です。

対象とする行列は、一般行列、バンド行列、正定値行列、対称行列、エルミート行列、三重対角行列であり、解法としては、コレスキー分解、直交三角分解、特異値分解が用意されています。

この PDSLIB/SX に含まれている LINPACK(以下、NEC 版 LINPACK と呼ぶ)は、Argonne

National Laboratory の Dr. Jack Dongarra、カリフォルニア大の Dr. Jim Bunch、National Laboratory の Dr. Jack Dongarra、カリフォルニア大の Dr. Jim Bunch、ニューメキシコ大の Dr. Cleve Moler、メリーランド大の Dr. Pete Stewart によって開発されたオリジナル版の LINPACK をベースに

SX-4 上に移植したものです。

4. 7. EISPACK

EISPACK は、9 種類の行列の固有値と固有ベクトルを計算する FORTRAN サブルーチン群です。

EISPACK は、複素一般行列、複素エルミート行列、実一般行列、実対称行列、実対称バンド行列、実対称三重対角行列、特殊実三重対角行列の固有値問題、及び実一般行列、実対称行列の一般化固有値問題を解くことができます。さらに、特異値分解を用いて、ある種の最小二乗問題を解く 2 つのサブルーチンが用意されています。

5. プログラミング支援環境

SX-4 では、様々なプログラム性能向上支援ツールをサポートしています。

5. 1. ANALYZER-P/SX

ANALYZER-P/SX は、FORTRAN77 プログラムのベクトル化および並列化に関するチューニングのための静的／動的特性を解析する性能向上支援ツールです。解析で得られた情報は、リスト形式で出力される他、解析情報データベースに登録することもできます。登録された情報は、PARALLELIZER/SX から対話的に編集表示することもできます。

ANALYZER-P/SX は fanp で利用できます。詳しくは man コマンドを活用してください。

【例】 % fanp -tm -am -AO do -multi test1.f test2.f

** ANALYZER-P/SX is started.

<step1> Collecting vectorization/parallelization information.

<step2> Pre-Processing.

<step3> Generating object code.

<step4> Generating library source code.

<step5> Generating executable object code.

<step6> Executing.

Path measurement

DO loops are analyzed

Analyzing information

CPU time

Instruction count

Memory access

Microtask residence time

<step7> Post-Processing.

** ANALYZER-P/SX is ended.

解析する主な情報は、以下の通りです。

(a) 静的解析情報

- プログラム構成情報（木構成により表現したもの）
- プログラム単位間の呼び出しに関する相互参照情報プログラム全体
- 仮引数と実引数の対応情報
- 共通ブロックの要素に関する相互参照情報
- 同期／排他制御に関する情報
- マルチタスクの不正な制御に対する警告メッセージプログラム単位
- 英字名の相互参照

(b) 動的解析情報

プログラム全体

- 実行開始からの経過時間、CPU 時間
- MOPS 値、MFLOPS 値
- バンク競合の発生状況
- ベクトル化率（ベクトル演算率）、平均ベクトル長
- 並列化率、平均プロセッサ台数
- 実行開始からの経過時間、CPU 時間

マクロタスク

- バンク競合の発生状況
- ベクトル化率（ベクトル演算率）、平均ベクトル長
- 同期待ち時間
- 各プログラム単位ごとの実行回数、CPU 時間
- ベクトル化率（ベクトル演算率）、平均ベクトル長
- CALL 文ごとの実行回数、実行時間、バンク競合の発生状況
- 利用者指定範囲の実行回数、実行時間、バンク競合の発生状況

プログラム単位

- DO ループごとの実行回数、実行時間、バンク競合の発生状況
- 各文の実行回数とコスト
- 条件付き分岐先ごとの選択回数とその比率
- ベクトル化された DO ループの表示

5.2. ANALYZER90/SX

ANALYZER90/SX は、Fortran90 で記述されたプログラムの動的特性を解析する性能向上支援ツールです。ANALYZER90/SX は、ANALYZER-P/SX で測定できたプログラム単位毎の実行情報を内部副プログラムやモジュール副プログラム毎にも解析することができます。また、配列演算に対してもベクトル化最適化などの情報に加えて従来 ANALYZER-P/SX で解析できた DO ループの実行情報と同等の実行情報を解析することができます。その他、ANALYZER90/SX は、SX-4 上の実行ステップ以外をワークステーション上で行うクロス環境機能を備えています。

ANALYZER90/SX は、fan90 で利用できます。詳しくは man コマンド又は www のオンラインマニュアルを活用してください。

【例】 % fan90 -am test1.f

```
fan90 is started.  
** Compiling analysis program.  
Parsing.  
Generating analysis program source file  
Generating analysis program object file.  
** Linking analysis program.  
Generating runtime library source file.  
Generating analysis program object module file.  
** Measuring execution information.  
Analyzing information  
    CPU time  
    Instruction count  
    Memory access  
** Analyzing execution information.  
fan90 is terminated.
```

プログラム全体、手続き単位、ループ単位、配列演算単位に解析する主な情報は、以下の通りです。

- 実行開始からの CPU 時間
- MOPS 値、MFLOPS 値
- バンク競合、キャッシュミスの発生状況
- ベクトル演算率、平均ベクトル長
- 平均ループ長
- ベクトル化最適化情報

5.3. PARALLELIZER/SX

PARALLELIZER/SX は、ANALYZER-P/SX で出力した静的解析情報や動的解析情報および ANALYZER90/SX で出力した動的解析情報をもとに、プログラムのベクトル化の状況を把握するために必要な情報を表示したり、ベクトル化を促進するためにプログラムを修正する作業をワークステーション上で対話形式に進めていくツールです。静的解析は PARALLELIZER/SX の中から実行することもできます。また、ユーザインタフェースには X ウィンドウを使用して、グラフなどによるわかりやすい表示と簡単な操作で行なえます。詳しくは man コマンド又は WWW オンラインマニュアルを活用してください。

【例】 % setenv DISPLAY xxxxx:0.0
% xpara

PARALLELIZER/SX の主な機能は、以下の通りです。

情報の表示

- 一覧表示 タスク単位、プログラム単位、ループ単位の実行時間

- 英字名情報表示
- プログラム構造の経路図化情報の表示
- 動的解析情報のグラフ化チューニング前後の比較結果のグラフ化
- プログラム単位の原始プログラム表示
- ベクトル化や並列化の可否および不可理由、阻害 要因の表示

変数トレース

- 変数の使用状況を経路図に表示

プログラムの編集・変換

- プログラム単位のエディタ編集

プログラムの解析

- 静的解析

5.4. C-ANALYZER/SX

C-ANALYZER/SX は、C プログラムのチューニングをサポートする性能支援ツールです。利用者は C プログラムのソースファイルの全部または一部分を本ツールを通して翻訳／実行することによって、各種の編集された実行時情報を得ることができます。特に、ループに対する解析結果はベクトル化の為のチューニング情報として、利用者によるきめ細かなチューニングを支援します。詳しくは man コマンド又は WWW オンラインマニュアルを活用してください。

【例】 %ca [オプションの並び] ファイル名の並び

C-ANALYZER/SX が出力する主な情報は、以下の通りです。

プログラム全体

- 実行開始からの経過時間、CPU 時間
- MOPS 値、MFLOPS 値
- バンク競合の発生状況
- ベクトル化率（ベクトル演算率）、平均ベクトル長
- 並列化率、平均プロセッサ台数

関数単位

- 各関数単位ごとの実行回数、CPU 時間、MOPS 値、MFLOPS 値、バンク競合の発生状況、ベクトル化率、（ベクトル演算率）、平均ベクトル長
- 関数呼び出しごとの実行回数、CPU 時間、バンク競合の発生状況
- ループごとの実行回数、CPU 時間、バンク競合の発生状況
- 利用者指定範囲ごとの実行回数、CPU 時間、バンク競合の発生状況
- 各文の実行回数とコスト
- 条件付き分岐ごとの選択回数とその比率

5.5. prof（プロファイラ）

prof コマンドは、実行時プロファイル・ファイルを解釈し、解析情報を編集出力します。実行時プロファイル・ファイルは翻訳時のオプションを指定により採取することができます。詳しくは man コマン

ドを活用してください。

【例】 `% f77sx prof.f -p`

`% a.out`

プログラムを実行すると `man.out` のファイルが作成され、このファイルが `prof` の入力ファイルになります。

`% prof`

`prof` により得られる解析情報には、以下の様なものがあります。

- 関数の呼び出し回数
- 関数の呼び出し時間
- 実行に要した時間（割合）のヒストグラム
- 各種コマンド

6. プログラム開発支援

SX-4 は、利用者のプログラム開発を支援するための様々なシステムを備えています。

6.1. SCCS

ソースコード管理システム SCCS(Source Code Control System) は、ソフトウェア開発においてソーステキストの変更修正管理を支援する目的で作られたソフトウェア支援ツールです。このシステムには、モジュールのすべてのバージョンの蓄積、改訂、検索機能、改定権の管理機能、バージョン番号によるロードモジュールの識別機能、各ソフトウェアの変更の実施者名、変更日付、変更理由の記録機能などがあります。

6.2. lint

lint は C 言語で記述されたソースファイルを解析し、バグや曖昧な文の検出をします。lint が行うソース解析は、C コンパイラ以上に厳密に行います。そのため、構文上正しい文に対しても無駄な行や不正な動作を引き起こしそうな文についても診断を与えてくれます。また、異なる機種種のコンピュータおよびオペレーティングシステムに移植する際に問題となるポータビリティに関する制限を検出するときにも利用できます。

主な機能は、次の通りです。これらにより、コンパイルエラーや他のマシンへの移植上のチェックが行えます。

- 未使用変数や関数のチェック
- 値のセットされた変数や参照されない変数のチェック
- 実行されない文のチェック
- 関数のリターンのチェック
- 構造体、変数などの代入文に対する型のチェック
- キャストのチェック

- ノンポータブルキャラクタのチェック
- SUPER-UX ANSI C仕様チェック

7. デバッグ支援

SUPER-UX では、プログラム開発を支援するための強力なデバッグ支援機能として sdb の他、dbx をサポートしています。dbx は、主要言語である FORTRAN77,90 および C 等に対応したデバッグ機能と使い易いユーザインタフェースを備えています。

【例】 dbx prog

dbx の基本機能としては、以下の様なものがあります。

- オリジナルソースレベルでシンボリックにデバッグすることができます。
- これは C 言語で記述されたプログラム、FORTRAN 言語で記述されたプログラム、これらの混在するプログラムに対しても可能です。
- アセンブラで記述されたプログラムに対しても機械語レベルでデバッグすることができます。
- 実行文にブレークポイントを設定し、実行追跡やブレークポイントからの実行再開ができます。
- ブレークポイントでの関数、パラメータ調査ができます。
- デバッグの途中でソース行を表示することができます。

さらに拡張機能として、以下のデータ等を扱うことができます。

- 8バイトの整数データ
- 8バイトまでの論理型データ
- 4倍精度までの浮動小数点
- 4倍精度までの複素数型データ
- 標準浮動小数点形式と拡張浮動小数点形式のデータ
- 7次元までの配列
- 整合配列・疑寸法仮配列などの FORTRAN 言語特有の配列

また、並列処理プログラムのデバッガとして pdbx があります。pdbx により FORTRAN 言語および C 言語で書かれた並列処理を行うプログラムのデバッグも行うことができます。

pdbx は dbx の持つ全ての機能に加えて、次の機能が追加されています。

- ブレークポイントで他のタスクの状態を調べることができます。
- ブレークポイントに到達したタスクとは異なるタスクに含まれるデータの内容を参照変更することが可能です。
- デッドロック検出時にデバックすることが可能です。

8. エディタ

SX-4/64M2 では、以下のエディタが利用できます。初心者から上級者まで、また端末や用途に応じて選択することができます。

○行エディタ `ed, sed`

○画面エディタ `vi, Emacs`

いずれのエディタにおいても、ワークステーションでの UNIX ユーザには広く親しまれているものであり、特別な知識を修得することなく利用することができます。

さらに、日本語機能をサポートしています。画面エディタ `vi`、`Emacs` においても日本語を利用することができます。

[通常で端末利用]

【例】 `% emacs -nw`

[X-window system で利用]

○SX-4/64M2 の `emacs` を日本電気以外のワークステーションに `DISPLAY` 指定で利用する場合、`emacs` で使用するフォントを指定する必要があります。

【例】 `% emacs -font r16 -kfont rk16 -display xxxxxxx:0.0`

xxxxxxx は ワークステーションの `ip` アドレスです。

○ワークステーション上のホームディレクトリ直下の `.Xresources` ファイルに以下の 2 行を追加して利用する方法

`emacs*Font: r16`

`emacs*KanjiFont: rk16`

9. グラフィック

9.1. SXview/IMG

科学技術計算により得られた数値データを可視化し、結果の理解を容易にするビジュアルシミュレーションシステムです。本システムは、SX 上で高速に画像を生成し、フレームバッファに出力します。またイメージファイルに格納した画像をフレームバッファに出力することにより、リアルタイムアニメーションが可能です。

9.2. SXview/GWS

科学技術計算により得られた数値データを可視化し、結果の理解を容易にするビジュアルシミュレーションシステムです。グラフィックワークステーション上への可視化表示、VTR コマ撮りによるアニメーション作成等ができます。クライアント・サーバ型の分散処理システムであり、GUI 操作による各種図形処理ができます。

9.3. AVS

AVS は、プログラミング不要な可視化ツールです。個別の可視化プロセスに対応した各種のモジュール

ルを、データの流れに従って組み合わせるだけで、自由自在に可視化ができます。ネットワークエディタを用いて、アイコン化されたモジュールをグラフィカルに操作できるため、プログラムの作成や編集は不要です。8ミリVTRのコマ撮りによるアニメーション作成等ができます。

【例】 設定ファイル /usr/avs/runtime/avsrc を ホームディレクトリに .avsrc としてコピーしてください。

```
% setenv DISPLAY xxxxx:0.0
```

```
% avs
```

9.4. グラフィック基本システム—GKS

Graphical Kernel System(GKS)は、装置独立な二次元の基本的図形入出力機能を提供するサブルーチンライブラリです。

本 GKS は、ISO(国際標準化機構)におけるコンピュータグラフィックスの国際標準(ISO7942)に準拠しており、以下のような特長を有しています。

- ISO7942 で定められたレベル 2C 機能のサポート

- FORTRAN77 および C 言語インタフェースのサポート

- (ISO/IEC 8651/1 GKS Language bindings-FORTRAN)

- 多様なグラフィックス装置のサポート

- 応用プログラムの高い可搬性

9.5. 3次元グラフィックスシステム—PHIGS PLUS

PHIGS PLUS は、国際標準(ISO/IEC9592)の PHIGS(Programmer's Hierarchical Interactive Graphics System)および標準化作業中(DP9592-4)の PHIGS PLUS に準拠した3次元グラフィックスシステムです。

PHIGS PLUS では、C および FORTRAN77/SX との言語インタフェースを提供しており、X ウィンドウに図形の表示を行うことができます。応用プログラムから本ライブラリの関数またはサブルーチンを直接呼び出すことによって、実際に図形を表示するハードウェアの詳細を考慮することなく、論理的なインタフェースで三次元の図形の描画、入力、制御を行うことができます。

10. アプリケーション

SX-4 のアプリケーションとして、構造解析・流体解析・計算化学・電磁場・電子回路解析などがあります。また、これらのプリ・ポスト処理を行なうアプリケーション JVISION、CERIUS2、MENTAT、MSC/PATRAN、ICEM/CFD、Pre・M、ATRAC などがあります。プリ・ポスト処理は ccindigo[01,d01] でお使いください。

これらのアプリケーションは下記のディレクトリ配下に納められています必要に応じて、各アプリケーションの実行ディレクトリまでのパスを追加して利用してください。

```
[ SX-4 ] % set path = ($path /usr/local/appli/)
```

```
[indigo] % set path = ($path /usr/local/appli/)
```


また、プリ・ポスト処理は ccindigo[01,d01]の X-window 上で動作しますので他のワークステーションから利用する場合 DISPLAY で画面を張りつけください。

【例】 % setenv DISPLAY xxxxx:0.0

[MARC の場合]

【例 SX-4】 % set path = (\$path /usr/local/appli/marck62/bin)
% marck62 -option

10.1. 構造解析

10.1.1. FEMLAB/WORKS/SX (プリ・ポストシステムは JVISION)

有限要素法を用いた構造解析プログラムであり、線形構造解析を高速かつ高精度に求解するアルゴリズムを採用しています。静解析、固有モード解析、座屈解析、周波数応答解析、熱伝導解析、簡易最適設計などが可能です。

また、簡易言語による解析処理の制御が可能であり、ユーザ固有の解析システムの構築ができます。

下記コンポーネントから構成されます。

○FEMLAB/SX (構造解析ソルバ)

【例 SX-4】 % set path = (\$path /usr/local/appli/femlab/bin)
% femlab < 入力データ > 出力データ

○JVISION (形状／格子生成、解析条件設定、解析結果の可視化)

【例 indigo】 % set path = (\$path /usr/local/appli/jvision/bin)
% jvgo

10.1.2. MARC (プリ・ポストシステムは MENTAT)

MARC は、変位法に基づく有限要素法の構造解析プログラムであり、特に、非線形分野の解析機能に優れています。線形、大変形、座屈、弾塑性、動的非線形、破壊、境界非線形、熱伝導、連成解析 (固体と流体、熱と応力、電気伝導と熱伝導) などの各種解析が可能です。

【例 SX-4】 % set path = (\$path /usr/local/appli/marck62/bin)
% marck62 -option

【例 indigo】 % set path = (\$path /usr/local/appli/mentat231/bin)
% mentat

10.1.3. LS-DYNA3D (プリ・ポストシステムは JVISION)

LS-DYNA3D は、陽解法を用いた高度な解析手法により、優れた計算効率を実現します。接触問題の取扱いが容易で、材料モデルが豊富なので、構造物の衝突や落下の挙動を現実には極めて近く再現できます。自動車の衝突解析や核燃料容器の落下解析のほか、従来のプログラムでは解析が困難だった成形加工のシミュレーションにも適用できます。

【例 SX-4】 % set path = (\$path /usr/local/appli/JRI/ls936/bin)

```
% lsdyna i=data
```

10.1.4. MSC/NASTRAN (プリ・ポストシステムは MSC/PATRAN)

MSC/NASTRAN は、線形・非線形静解析、固有値解析、線形・非線形動的解析、熱伝導解析、空力弾性解析、音響解析等の解析が行えます。また、多重レベルの部分構造解析、モード合成解析、周期対称法による解析、感度解析、複合材料、流体・構造連成解析等の手法により、さらに汎用性が広がります。さらに、各種解析において、数多くの解析手法の選択が可能であり、幅広いアプリケーションに対応できます。

【例 SX-4】 % set path = (\$path /usr/local/appli/NAST6822/bin)
% nast682 data_file [options]

10.2. 流体解析

10.2.1. α -FLOW/SX (プリ・ポスト機能を包含。ただしプリの一部は ICEM/CFD)

有限差分法を用いた流体解析プログラムであり、以下のコンポーネントから構成されます。

- α -FLOW/SX/BF～非圧縮性流体に関する流れ場や熱輸送の解析
- α -FLOW/SX/CF～圧縮性流体に関する流れ場の解析
- α -FLOW/SX/CR～燃焼・化学反応を伴う流れに関する解析
- α -FLOW/SX/FS～自由表面を伴う流れに関する解析
- α -FLOW/SX/CR～燃焼・化学反応を伴う流れに関する解析
- α -FLOW/SX/FS～自由表面を伴う流れに関する解析
- α -FLOW/SX/VS～入力データの作成 (プリ処理) の一部および解析結果の可視化 (ポスト処理)

プリ処理の中で形状生成および格子生成には ICEM/CFD を利用します。

また、ポスト処理の一形態として、解析の実行と同時に可視化を行うリアルタイムビジュアライゼーション機能 (現時点では /BF、/CF のみ対応) が利用できます。

【例 SX-4】 環境設定
% setenv ALFAHOME /usr/local/appli/AlpaFlow

```
% set path = ($path $ALFAHOME)  
計算モデルによって実行モジュール(felbfc,felcrt,galaxy,hiants,icarus)が異なるので /usr/local/appli/AlpaFlow/sample 配下の各モデルのシェルスクリプトを参照してください
```

ICEM/CFD のアプリケーション

【例 indigo】 % setenv ICEM_ACN /usr2/local/appli/icem3.32
% set path = (\$path /icem/bin)
% icemcfd

10.2.2. STREAM (プリ・ポストシステムは Pre・M, ATRAC)

有限体積法による3次元汎用熱流体解析プログラムです。層流・乱流、ニュートン／非ニュートン流体、自由表面機能／多種流体、固体と流体の温度連成、定常・非定常／圧縮性・非圧縮性、拡散物質の

温度分布解析が可能です。

```
【例 SX-4】 % set path = ( $path /usr/local/appli/Dcradle/bin)
              % stream29
```

Pre・M と ATRAC アプリケーション

```
【例 indigo】 % set path = ( $path /usr/local/appli/Dcradle/bin)
               % perm16 または % atrac
```

10.2.3. PHOENICS-2 (プリ・ポスト機能を包含)

PHOENICS-2 では、次の解析が可能です。

- 多相流：(固体、液体、気体) 3相流可能 -ムービング BFC
- 伝熱：対流、輻射、対流熱伝達／固体内熱伝導 -乱流モデル
- 温度による浮力、任意形状の表現 -パーティクルのトラッキング
- 定常・非定常、圧縮性・非圧縮性流体
- 非ニュートン流体：ビンガム流体、ベキ乗則
- 多孔質内の流れ、抗力／揚力の計算、旋回流
- 自由表面 /2 流体の界面

```
【例 SX-4】 % set path = ( $path /usr/local/appli/CHAM)
              PHOENICS コマンド
```

10.3. 計算化学

10.3.1. AMOSS/SX

分子の物理的・化学的性質および分子の安定構造などを、非経験的分子軌道法に基づく数値計算により、理論的に解析・予測する分子科学プログラムです。

電気多重極子モーメント、ポピュレーション解析、静電ポテンシャルなど、多種類の物理量や解析情報が高精度で得られます。また、最新の数値アルゴリズムの採用により比較的大きな分子に対しても、現実的な時間内での解析が可能です。

ワークステーション上で稼働するプリ・ポストシステム (AVS を利用) も統合されており、入力データの作成や解析結果の可視化も可能です。

———— AMOSS/SX の NQS を使った並列化時の CPU 台数、メモリについての説明 ————

[並列実行可能性]

AMOSS/SX は、シングルノード内(最大 32CPU)での並列実行が可能です。

[並列実行の方法]

並列で実行させるには、環境変数 AMOSS_PARALLEL に使用する CPU 台数を指定します。

例えば、2 つの CPU で並列実行させるには以下のように指定します。

[バッチリクエスト用の使用例]

C シェルの場合

```
#!/usr/bin/csh
```

```
#@$-q p4          ジョブクラス p4 を指定する
```

```

#@$
#
setenv AMOSS_PARALLEL 4
setenv F_UFMTIEEE 60
setenv F_UFMTADJUST NO
setenv LM /usr/local/appli/amos/amoss
setenv DAT ~/atxt/hf.d
setenv OUT ~/atxt/hf.out
$LM <$DAT>$OUT

```

[NQS のキュー設定について]

並列化されているモジュールの並列効率、2CPU で約 1.6～1.9 です。

計算内容や規模などに依存しバラつきがあるため、NQS での CPU 台数の設定は、2～8 台を推奨します。

[メモリサイズ]

並列に実行させて経過時間を短縮させたいと考えるのは、その多くは大規模な計算を行う場合であると思われます。

このような大規模な計算の場合、AMOSS/SX は実行時のメモリサイズをより多く要求します。

また、AMOSS_PARALLEL に指定する値が大きいほどさらに多くのメモリを要求します。

AMOSS/SX のメモリサイズの既定値は約 130MB (32CPU 実行時)です。#1cpu 時のメモリ既定値は、約 90MB です。

これは、環境変数 AMOSS_MEM により変更可能であり、最大 2GB まで指定できます。したがって、NQS でのメモリサイズの設定は、512MB～2GB を推奨します。インタラクティブの場合も NQS の利用と同等です。

10.3.2. AMBER

AMBER は、巨大分子シミュレーションパッケージプログラムであり、エネルギー最適化計算、基準振動モード計算、分子動力学計算、自由エネルギー摂動計算、分子力場計算、エネルギー極小化計算が行えます。

【例 SX-4】 % set path = (\$path /usr/local/appli/amber/a41nec/exe)

計算モデルによって実行モジュールが異なります

詳細は /usr/local/appli/amber/a41nec/doc 配下の intoro.doc,tutorial.doc を参照してください

10.3.3. MOPAC (プリ・ポストシステムは Cerius2)

半経験ハミルトニアンとしては、MNDO、MINDO/3、AM1 および PM3 が備わっており、中性分子、ラジカル、イオンおよびポリマに関して、振動スペクトル、熱力学的物理量、同位体置換効果およ

び結合に関する力の定数を計算します。

```
【例 SX-4】 % set path = ( $path /usr/local/appli/mopac/mopac7)
              % runmopac7
```

```
【例 indigo】 % set path = ( $path /usr/local/appli/cerius2_2.0)
               % cerius2
```

10.3.4. GAUSSIAN (プリ・ポストシステムは Cerius2)

GAUSSIAN は、分子構造、反応の遷移状態および数多くの分子物性値を計算することができます。本パッケージでは、s、p、d および f 軌道に対して、短縮ガウス型関数を使用した 1 電子、2 電子積分計算機能、制限・非制限 Hartree-Fock 波動関数に対する SCF 計算機能、MP 摂動法、配置間相互作用法、coupled-cluster 法を使用した相関エネルギー計算機能を含んでいます。エネルギー極小点や鞍点を自動的に決定し、エネルギー勾配やヘシアンを解析的あるいは数值的に計算することができます。

[利用方法]

SX-4 の .login ファイルの最後に次の 2 行を追加して下さい

```
setenv g94root /usr/local/appli
source $g94root/g94/bsd/g94.login
```

【例】入力データは h2o.01.com、入力スクリプトを h2o.gaussian とすると

```
#!/usr/bin/csh
#@$-q p4
#@$-o h2o.01.stdout
#@$-e h2o.01.stderr
#@$
# run script for gaussian
#
setenv F_PRINTFDIFF NO
setenv F_PROGINF DETAIL
setenv F_ERROPT1 271,271,2,1,2,2,2
setenv GAUSS_EXEDIR /usr/local/appli/g94
${GAUSS_EXEDIR}/g94 < h2o.01.com
```

[実行方法] % ccsx40% qsub h2o.gaussian

gaussian は、計算の途中に大量のテンポラリファイルを使います。テンポラリファイル領域として /jtmp の配下を利用することをお勧めします。/jtmp 領域を利用するには、下記の 2 行を #run script for gaussian の以降に追加して下さい。

ただし、/jtmp 配下のファイルはジョブの終了時に解放されます。

[一時ファイルを利用する場合]

```
% cp h2o.01.com /jtmp/.
```

```
% cd /jtmp
```

10.3.5. GAMESS

GAMESS (General Atomic and Molecular Electronic Structure System) は、非経験的分子軌道計算プログラムであり、多種多様な波動関数と基底関数により、分子の化学的特性の計算を行います。

```
【例 SX-4】 % set path = ( $path /usr/local/appli/gamess)
              % gamess.00.x
```

10.4. 電磁場・電子回路解析

10.4.1. PLANC-MM (プリ・ポストシステムを包含)

- モーメント法による厳密な解析
- 任意形状の導体が解析可能
- アンテナの解析、電磁波散乱の解析、EMC 評価、電磁界分布の計算
- 線状で数 100 λ 、面状で数 λ の大きさまでの導体が計算可能

```
【例 indigo】 % set path = ( $path /usr/local/appli/planc/mm)
```

実行ディレクトリに inp.dat, sol.dat, post.dat, pre.dat をコピーしてください

```
% mm[1-3] %p2 %p3 %pw
```

10.4.2. PLANC-FDTD (プリ・ポストシステムを包含)

- 時間領域での MAXWELL 方程式の差分法による解析
- 任意形状、任意物性値の誘電体、導体の解析
- Fourier 変換による周波数領域の解析
- 近傍界、遠方界の解析
- 10-20 λ までの大きさの対象が計算可能

```
【例 indigo】 % set path = ( $path /usr/local/appli/planc/fdtd)
```

実行ディレクトリに inp.dat, post.dat をコピーしてください

```
% fd[1-3] %p2 %p3 %pw
```

10.4.3. PLANC-MSA (プリ・ポストシステムを包含)

- マイクロストリップアンテナの解析
- Sommerfeld 積分とモーメント法による厳密な解析
- 放射パターン、入力インピーダンスの計算

```
【例 indigo】 % set path = ( $path /usr/local/appli/planc/msa)
```

実行ディレクトリに inp.dat, sol.dat, post.dat, pre.dat をコピーしてください

```
% ms[1-3] %p2 %p3
```

[プリ・ポストシステムの使用方法]

```
【例 indigo】 % set path = ( $path /usr/local/appli/planc/pcal/bin) を追加する
```

10.4.4. SPICE

SPICE(Simulation Program with Integrated Circuit Emphasis)は、非線形直流解析、非線形過渡解析と微小信号線形交流解析のための回路シミュレーションとしてパークレイのカリフォルニア大学で開発されました。

現バージョンの SPICE3F4 で扱えるのは、抵抗、キャパシタンス(コンデンサ)、インダクタンス(コイル)、相互インダクタンス、伝送ライン、独立電圧(電流)源、制御電源、制御スイッチと半導体素子：ダイオード、バイポーラトランジスタ(BJT)、JFET、GaAsFET、MOSFET、などです。

```
【例 SX-4】 % set path = ( $path /usr/local/appli/spice)
               % spice3
```

10.4.5. EGS4

EGS(Electron Gamma Shower)4 コードは、SLAC(Stanford Linear Accelerator Center)において開発されてきた電磁カスケードシミュレーション用の汎用 MC コードです。ユーザは提供されているライブラリを使用して、アプリケーションを構築する事が可能です。

```
【例 SX-4】 % set path = ( $path /usr/local/appli/egs4)
               % eqs4_run
```

11. ワークステーションのソフトウェア

高度なグラフィックス処理等、主システムだけでは提供が難しい機能を補うために、UNIX サーバとワークステーションが設置されています。また、画像処理に関するソフトウェアとして可視化機能のアプリケーション avs 及び Explorer があります。利用者は IRIS Explorer 及び avs に附属する各種モジュールを組み合わせるだけで、専門的なプログラミング知識がなくても可視化を短時間で行うことができます。また、これらのソフトウェアで取り扱う画像（拡張子.x .rgb .sgi 等）をアニメーション化することも可能です。そのための画像変換ツールをはじめ表示ソフト等も紹介します。

表 11.1. 言語・支援ツール

	calpha01	cconyx01	ccews01,03	ccindigo01,d01
FORTRAN77	○	○	○	○
power FORTRAN	×	○	×	×
Analyzer	×	○	×	×
c	○	○	○	○
power c	×	○	×	×
c++	○	○	○	○
Common Lisp	○	×	○	×
UTILISP	×	×	○	×
Fanalyzer	×	×	○	×
pascal	×	×	○	×
Allegro	×	×	○	×

PROLOG2		×	○	×
---------	--	---	---	---

なお、FORTRAN90 は ccup02 でサービスしております。

表 11.2. ワークステーションのアプリケーション

システム	アプリケーション
cconyx01	EXPLORER : 画像表示ソフトウェア AVS : 画像表示ソフトウェア IRIS Inventor : マルチメディアプレゼンテーションツール SHOWCASE : 文書処理
ccindigo01 ccindigod01	AVS、EXPLORER : 画像表示ソフトウェア JVISION、CERIUS2、MENTAT、MSC/PATRAN、ICEM/CFD、Pre・M ATRAC などの流体解析・構造解析・計算化学・電磁場・電子回路解析の プリ・ポスト処理アプリケーション MAPLE V : 数式処理システム MOL-MOLIS : MOPAC インタフェースをフルサポートした、ビジュアル リゼーションツール
ccews01	GKS、XGKS : 図形処理 EASYDRAW II : 二次元簡易図形システム（プロッタ出力等を行う
ccews03	PIVOT : 日英及び英日翻訳支援ソフトウェア DWB、JWB : 文書処理 Image-Driver、Image View : イメージ支援処理 EXBASE、EXTOOL、EXWORK : AI micro Researcher II、SII : 統計解析 KERMIT,FTP : ファイル転送 FAX サービス : ワークステーションからの FAX への送出

11.3. imsl ライブラリの利用方法

imsl ライブラリは ccup02、ccindigo01 でお使いください。

[環境変数をセットします]

csh の場合 [ccindigo の場合] % source ~guest/imsl_indigo.csh

[ccup02 の場合] % source ~guest/imsl_up.csh

sh の場合 [ccindigo の場合] % source ~guest/imsl_indigo.sh

[ccup02 の場合] % source ~guest/imsl_up.sh

[fortran コンパイル・実行]

【例】 f77 -o 実行モジュール名 ソースファイル名 \$LINK_FNL

[オンラインドキュメント]

X-window 上で動作します **iptdoc** コマンドでオンラインメニューがでます。

11.4. mathematica の利用方法

ccindigo01 または cconyx01 でお使いください

X-window 上で動作します **math** コマンドで利用できます。

表 11.5. ワークステーションで利用できるソフトウェア

	calpha01	cconyx01	ccews01	ccindigo01
archie	○	×	×	×
bash	○	○	○	×
convert	○	×	×	×
diff	○	○	○	○
display	○	×	○	×
fgrep	○	○	○	○
find	○	○	○	○
gawk	○	○	○	×
gopher	○	×	×	×
ieditor	×	○	×	○
imake	×	○	○	○
jgawk	×	×	○	×
jLATEX	○	○	○	×
kermit	○	×	×	×
less	○	○	○	×
lha	○	○	○	×
netscape	○	○	×	×
NEmacs	×	×	○	×
mule	○	○	×	×
nkf	○	×	×	×
rn	○	×	×	×
tcsh	○	○	○	×
xv	×	○	×	×
xarchie	○	×	×	×
xcalender	×	×	○	×
xdiff	×	○	○	○
xedit	○	×	○	○
xgopher	×	×	○	×

11.6. ワークステーションの画像処理に関するソフトウェア

display (画像を見るソフト。ccalpha の X-window 上で動作)

【例】 `% display` 画像ファイル名 または、`% display` と入力し、マウスの左ボタンを押し、メニューから画像を選択する。

xv (画像を見るソフト。cconyx の X-window 上で動作)

【例】 `% xv` 画像ファイル名 または、`% xv` と入力し、メニューから画像を選択する。

convert (画像フォーマットの変換ソフト。ccalpha 上で動作)

display,xv 等に対応していないフォーマットも変換できます。

【例】 `% convert test.rgb test.ps`

RGB 形式の test.rgb を PS 形式の test.ps として変換する。

ghostview (PS 形式のファイルを画面上で見るソフト。ccalpha の X-window 上で動作)

PS 形式のファイルをプリンタで出力される場合はこれで確認できます。

【例】 `% ghostview` PS ファイル名 または、`% ghostview` と入力し、メニューから PS ファイルを選択する。

ファイルが複数ページとなる場合、PAGE をクリックしてページを進めたり、戻せたりできます。マウス中央のボタンでページ番号をクリックすれば、そのページが表示されます。また、ページをマークすることで、任意のページを印刷することができます。

snapshot (画面の画像を取り込むソフト。ccindigo,cconyx の X-windows 上で動作)

【例】 `% snapshot` と入力する。

画面左上のほうに snapshot と表示される。それにマウスのカーソルをあわせてマウスの左ボタンを押しながら画像を取り込む範囲まで枠を広げ、目的の場所にあわせる。その後、snapshot の位置にあわせ右クリックでセーブする。

showcase (画像の描写ソフト。ccindigo,cconyx の X-windows 上で動作)

【例】 `% showcase` と入力する。

出力ファイルとして ps ファイルも指定できます。

a2ps (ASCII ファイルを ps ファイルに変換するソフト。ccalpha 上で動作)

【例】 `% a2ps` ASCII ファイル > 出力ファイル

ASCII 形式の test.txt を PS 形式の test.ps として変換する。

dvi2kps,
jdvi2kps (TeX の dvi ファイルを ps ファイルに変換するソフト。 ccalpha 上で動作)
【例】 % **jdvi2ps** test.dvi > test.ps

dvi2tty (dvi ファイルをテキストファイルに変換するソフト。 ccalpha 上で動作)
【例】 % **dvi2tty** test.dvi > test.txt

11.7. プリンタに関するコマンド

lpr -Pacolor (乾式カラープリンタ装置。 ccalpha01,cconyx01,ccinidigo01 上で動作)
【例】 % **lpr -Pacolor** P S ファイル名

pictro SGI (湿式カラープリンタ (銀塩写真方式) 装置。 cciris01 上で動作)
【例】 % **pictro SGI** フォーマットファイル名
以下のフォーマットは指定することにより利用できます。
SUN,SUN_R,POINT,LINE,FACE,SGI,ALIAS,RLA,CDI,AVS,TIFF,
XWD
オプションの詳細は、**pictro -h** コマンドで確認してください。

12. 電子マニュアル

WWW でオンラインマニュアルを提供しています。

<http://www.center.osaka-u.ac.jp/center/cc-home-jp.html>

今のところ、SX-4 の概要、 α -FLOW/SX (流体解析システム) の操作説明、unix の利用の手引き、PPP 接続(Linux,Win95,Mac)の手引きなどを公開しております。逐次、充実していきます。

また、利用者専用のサービス (センターの ID が必要) として「SX-4 の日本語マニュアル SUPER-UX Release 7.1」を公開しております。なお、unix のオンラインマニュアル **man** コマンドを活用ください。

これ以外に冊子体マニュアルはメーカーより発行されております。これらの冊子体マニュアルは本館の図書資料室、講習会室、豊中データステーションに備えています。

13. 電子メールと電子ニュース

ccalpha01[133.1.4.22]はメール及びニュースサーバーも兼ねておりますので、大きな演算を行なうときは、ccup02、cconyx01 または、SX-4 などをご利用ください。メールを読むソフトとしてはエディタの emacs または mule を利用した、rmail,mh 及び mnews などを活用してください。

13.1. 電子メールの利用方法

【コマンド】 % **mule -e rmail**

以下に簡単な操作方法を示します。

C- は「[control] キーを押しながら指定キーを押す」を示します。

C-x C-c	終了
?	ヘルプの表示
	メールの先頭へ移動 (M-< と同じ)
SPC	メールをスクロール
DEL	メールをバックスクロール
n	次のメールへ移動
p	前のメールへ移動
<	最初のメールへ移動
>	最後のメールへ移動
j	数字の次に入力することにより、n 番目のメールに移動
d	このメールを不可視（表示しない）にして次に移動
C-d	このメールを不可視して前に移動
u	不可視メールの順次復活
e	メールの編集。 M-x rmail-cease-edit 編集終了
x	不可視メールの消去
s	不可視メールを消去後、ファイルを保存
q	メールを終了して、他のファイルの編集を行う
C-x C-s	ファイルを保存
g	サーバよりメールを受信する
m	送信メールを作成する (C-x 4 m と同じ)
C-x o	分割されたウィンドウの移動
C-x l	カーソルのあるウィンドウだけにする
C-c C-c	送信メールを送信する
c	前の送信メールを再編集する
r	返事のメールを作成する
f	メールを転送する
o	表示されているメールを追加保存する
C-o	表示されているメールを unix 形式で追加保存する
i	o や C-o で保存したファイルをメールとして扱う

13.2. 電子ニュースの利用方法

【コマンド】 % mnews -e

初めて起動する場合は多少の時間を必要とします。最初にグループが多数表示されますので、購読しないグループは u を入力して以降表示しないようにします。(その中に新しいグループが作成されると、また表示されます。)

電子ニュースのネチケット（ネットワークのエチケット）として以下の事に気をつけてください。

本当に投稿する必要があるのか。 そのニュースグループに適切であるか。

非難中傷や法律に触れていないか。 誤解を招いたり、内容に誤りはないか。

引用の量は適切か。表題は適切か。

他のコンピュータで利用できない外字や記号を使用していないか。

また、複数の記事を保存する場合は M（マルチマーク）を利用すると便利です。方法としては、保存したい複数の記事に M を入力し、マークを付けます。いずれかの記事で s を入力し、保存するファイル名、例えば kiji と入力すると順次 kiji01,kiji02,.....kijinn として保存されていきます。特に、実行形式のプログラム等を保存する場合に便利です。

【代表的なニュースグループとその内容】

bionet	生物学者向け
comp	コンピュータ・サイエンス関連
fj	日本語の記事が投稿される学術目的のグループ
gnu	FSF(Free Software Foundation)と GNU(GNU is Not Unix!)関連
jp	日本のインターネット関連
rec	趣味・芸術関連
sci	科学分野

【大阪大学ローカルのニュースグループとその内容】

center	阪大大型計算機センター NG
osaka-u	大阪大学ローカル NG

【ニュースのメニューとコマンド(詳しくはヘルプを参照してください)】

コマンド	各グループ(各グループ)	ニュース記事選択	ニュース記事表示
?	ヘルプ表示	〃	〃
p	前行に移動	前の未読記事に移動。	〃
n	次ぎに移動	次の未読記事に移動。	〃
b			上方の表示
i(SPACE)	選択する。	〃	下方 or 次記事の表示
q	抜ける。(終了)	〃	〃
/	前方検索。	〃	〃
¥	後方検索。	〃	〃
l	購読、未購読の表示切替。	既読、未読のの表示切替。	
u	購読、未購読を切替え。		
d		読んだ事にする。	
s		保存。	
M		マルチマークの指定/解除	
c		全記事を既読読んだ事にする	
g	新着記事を再チェック	〃	〃
<	先頭に移動	〃	〃
>	最後に移動	〃	〃
m	新たにメールを出す	〃	
a	新たに記事を投稿	〃	
r		記事に返信	〃

R	記事を引用して返信	〃
f	記事にフォロー	〃
F	記事を引用してフォロー	〃

14. インターネットサービス

POP、NNTP サービスでは特殊な UNIX コマンドを知らなくても "電子メール"、"電子ニュース" をパソコンで利用できます。また、PPP サービスは、自宅のパソコンなどから電話回線を用いて、これらの POP、NNTP サービスの他に WWW、FTP、Telnet などの各種インターネットサービスが利用できます。POP サービス・NNTP サービス・PPP (ダイヤルアップ IP 接続) サービスは「インターネットサービス」申請手続きをとることにより利用できます。

なお、通信速度は 28.8Kbps まで使用可能です。また、ISDN での受け口は用意しておりません。(ISDN の場合は TA のアナログポートにモデムを接続してご利用ください。)

- 【利用申請】 本センターの登録番号(例えば"a61234")を取得している方が利用申請を行えます。
- 【申請方法】 本センターの共同利用掛に用意してあります「インターネットサービス利用申請書」に必要事項を記入して提出して下さい。
- 【利用負担金】 インターネットサービス (POP, NNTP, PPP サービスをセットとして) の利用負担金は、1,000 円/月です。月の途中から申請されてもその月は 1,000 円を徴収いたします。

14.1. インターネットサービスを利用するための設定

【PPP で接続の設定項目】

PPP を使って通信するための設定は、各種の PPP クライアントソフトウェアによって設定項目は多少異なりますが、基本的には次に示すように設定してください。

(1) 電話番号 06-879-8983

(2) ユーザー名

本センターで取得している登録番号 (例えば"a61234") を設定します。

(3) パスワード

登録番号で使用している同じパスワードです。(大文字と小文字は別の文字として扱われます)

(注) ユーザー認証には PAP を使用しています。

(4) IP アドレス

パソコンの IP アドレスはサーバから自動的に割り当てられるように設定してください。

(5) サブネットマスク 255.255.255.0

(6) デフォルトゲートウェイアドレス 133.1.4.1

(7) ネーム (ドメイン、DNS) サーバのアドレス

133.1.119.1 133.1.181.1 133.1.192.4

14.2. POPサーバを使って電子メールを読むための基本設定

(1)電子メールアドレス

登録番号@center.osaka-u.ac.jp

【例】 a69999@center.osaka-u.ac.jp

(*)電子メールアドレスの別名登録

登録番号以外のメールアドレスも取得したい方は申請書に希望名を記述してください。

他の利用者と重複する場合は先着順となります。

別名（例えば氏名）@user.center.osaka-u.ac.jp

(2)POP メールサーバ名 pop-server.center.osaka-u.ac.jp [133.1.4.22]

14.3. NNTPサーバを使って電子メールを読むための基本設定

NNTP ニュースサーバ名 news-server.center.osaka-u.ac.jp

注意：ppp 接続以外は、利用する機器に割り当てられている IP アドレスをセンターに登録する必要があります。

15. FAXサービス

ccews01 上で FAX サービス（送信のみ）が利用できます。簡単なコマンドで文字及び TeX データを FAX に送信することができます（ただし、現在は国内送信のみ）。

以下、FAX への送信方法について説明します。（詳しくはログインメニューをご覧ください）

*文字データの送信の場合

【例】 % a2ps 送信ファイル | sendfax -n -m -t 3 -d 0-xxx-xxxx

*TeX データの送信の場合

【例】 % sendfax -n -m -t 3 -d 0-xxx-xxxx 送信ファイル

*1 バイト文字のみの ascii データ

【例】 % sendfax -n -m -t 3 -d 0-xxx-xxxx 送信ファイル

16. マルチメディア情報蓄積システム

マルチメディア情報処理システムとして、MPEG2 エンコーダ・デコーダシステム及びビデオオンデマンドサーバシステムが利用可能です。MPEG2 エンコーダにより映像情報を MPEG2 フォーマットで取り込み、ビデオオンデマンドサーバに貯えて MPEG2 デコーダにより取り出すことができます。また、ノンリニアビデオ編集システムでは映像情報をリアルタイムに取り込み、ノンリニアに編集することができます。編集の際にはハードウェア処理による様々な 3D エフェクトを用いることができます。更に、豊富な入出力機器、VCR、CD プレーヤ、イメージスキャナ、フルカラープリンタ、スイッチャ、DAT、オーディオミキサ等も利用できます。IRIS/Onyx の CG 出力などのビデオデータを効率的に編集・蓄積することができます。

16.1. ノンリニア編集システム

リニア編集システムと連携し、ビデオデータのノンリニアの加工・編集を行なうシステム (Avid Media Composer MC4000) です。パーソナルコンピュータ上のソフトウェアにより全ての操作を容易に行なうことができる、ノンリニア編集においてはリアルタイムで様々な 3D エフェクト処理が可能です。



16.2. MPEG2 エンコーダ・デコーダシステム

リアルタイムでビデオデータを取り込み、MPEG-1/2 フォーマットに圧縮するシステムと、VOD サーバーに蓄積された MPEG-1/2 フォーマットのビデオデータをイーサネットを通じてリアルタイムに展開して表示するパーソナルコンピュータです。

16.3. VOD サーバシステム

MPEG-1/2 フォーマットのビデオデータの共用利用を可能とするビデオサーバです。ビデオデータ編集ソフト (Star Works) により毎秒 30 フレームの動画再生が可能です。

16.4. A/B ロール編集システム

既存のビデオ処理システムに加えて VTR(PVW-2650)、エディットコントローラ (BVE-2000E) などのシステムを追加したりニア編集システムです。

