

# サイバーメディアセンター 大規模計算機システムの利用

大阪大学 情報推進部 情報基盤課

# 目次

- I. サイバーメディアセンターの大規模計算機システムのご紹介
- II. 大規模計算機システムへの接続
- III. コンパイル
- IV. ジョブスクリプト
- V. ジョブ投入

# I サイバーメディアセンターの 大規模計算機システムのご紹介

SX-9



SX-8R



NEC製のベクトル型スーパーコンピュータ  
2014年8月31日をもってサービス終了予定

# SX-ACE



NEC製のベクトル型スーパーコンピュータ  
2014年12月9日よりサービス開始予定

# 汎用コンクラスタ(HCC)



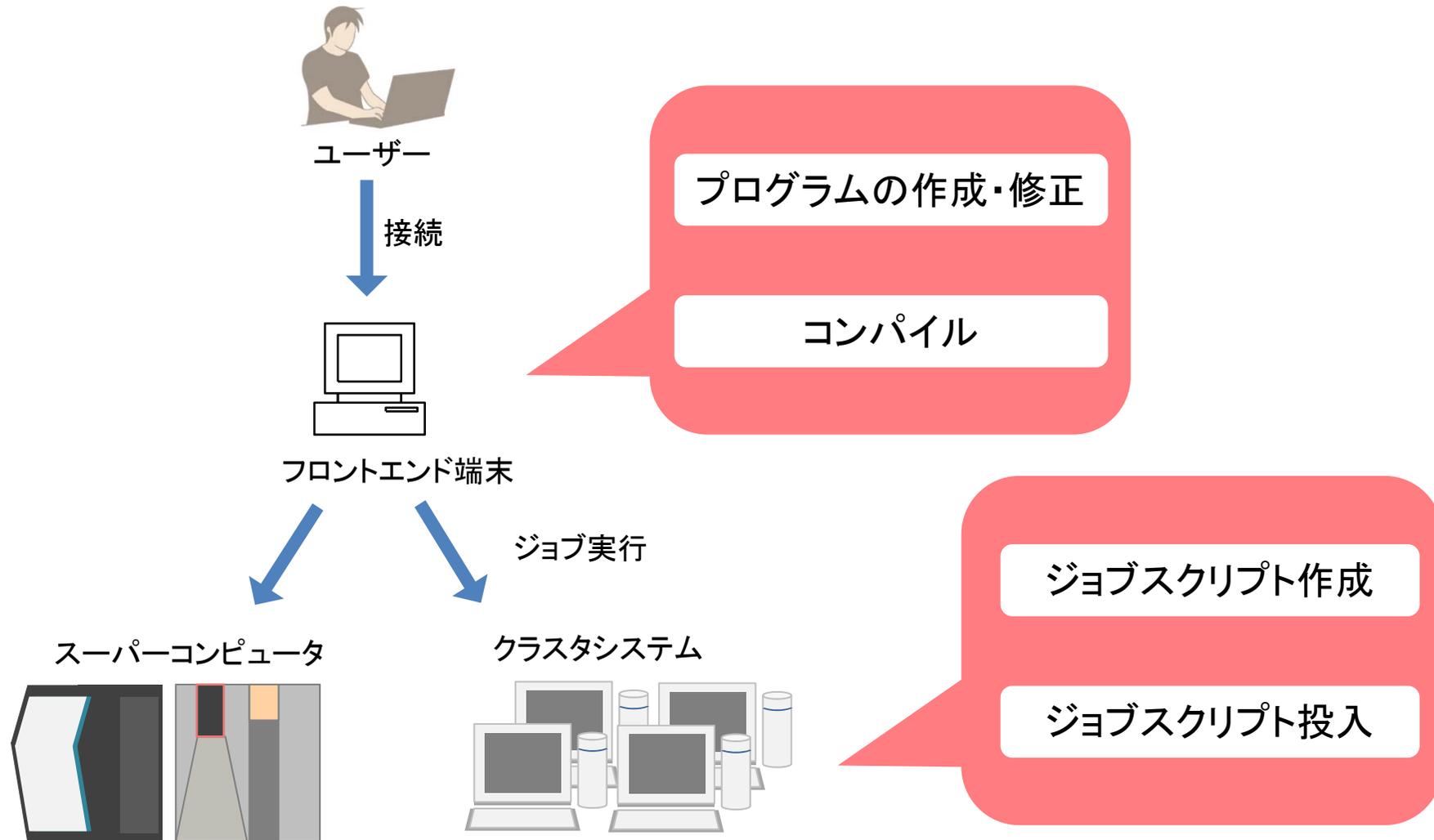
クラスタ計算機システム

汎用コンクラスタは学生用端末としても利用

## フロントエンド端末

- プログラムのコンパイルや計算結果の確認を行う  
作業用端末
- 基本的にユーザはフロントエンド端末にログインし、  
各計算機に対して、処理の実行を指示する
  - 基本的に、計算機本体へはログインしない
  - 処理の実行にはジョブスクリプトを使用(後述)

# 利用の流れ



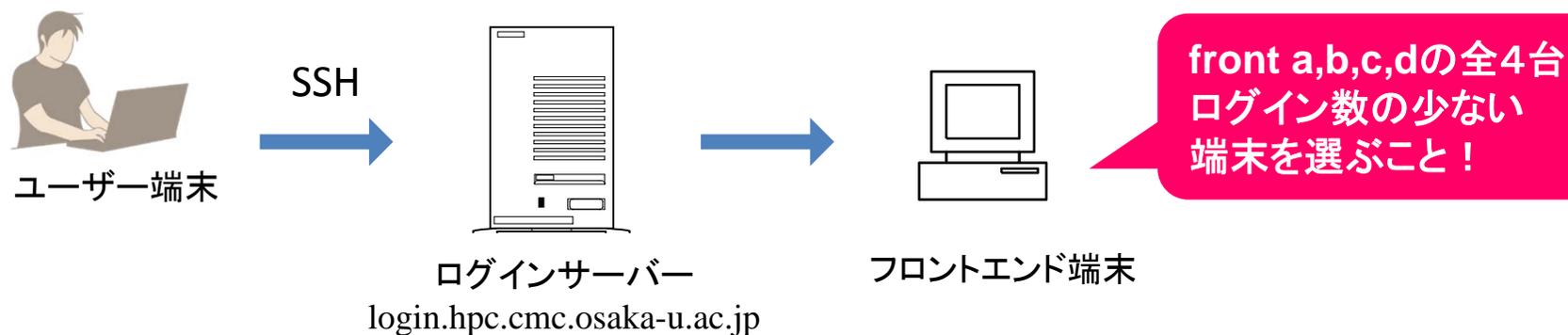
## Ⅱ 大規模計算機システムへの接続

# 大規模計算機システムへの接続

## ログインはSSH (Secure Shell) 接続

- 今回は「Tera Term」(SSHクライアント)を使用
- 接続先: **login.hpc.cmc.osaka-u.ac.jp**

参考: ファイル転送はSFTP接続 (接続先: **ftp.hpc.cmc.osaka-u.ac.jp**)



# 演習1(接続)

- フロントエンド端末へのログイン
  - Tera Term を利用した場合
  - 1. ホストに **login.hpc.cmc.osaka-u.ac.jp** を入力
  - 2. **ユーザ名とパスワード**を入力
  - 3. メニュー画面が表示
  - 4. [1]～[4] のどれかを選択

```
Cybermedia Center,Osaka University
Welcome to a large-scale computing system !!

Select the system number of the application

[0] The information from the center
[1] Frontend Terminal (fronta) >> online user(s) : 60
[2] Frontend Terminal (frontb) >> online user(s) : 6
[3] Frontend Terminal (frontc) >> online user(s) : 8
[4] Frontend Terminal (frontd) >> online user(s) : 7
[8] Super Computer (sx8)
[99] Logout (Session Close)

Enter number ? █
```

ログイン数

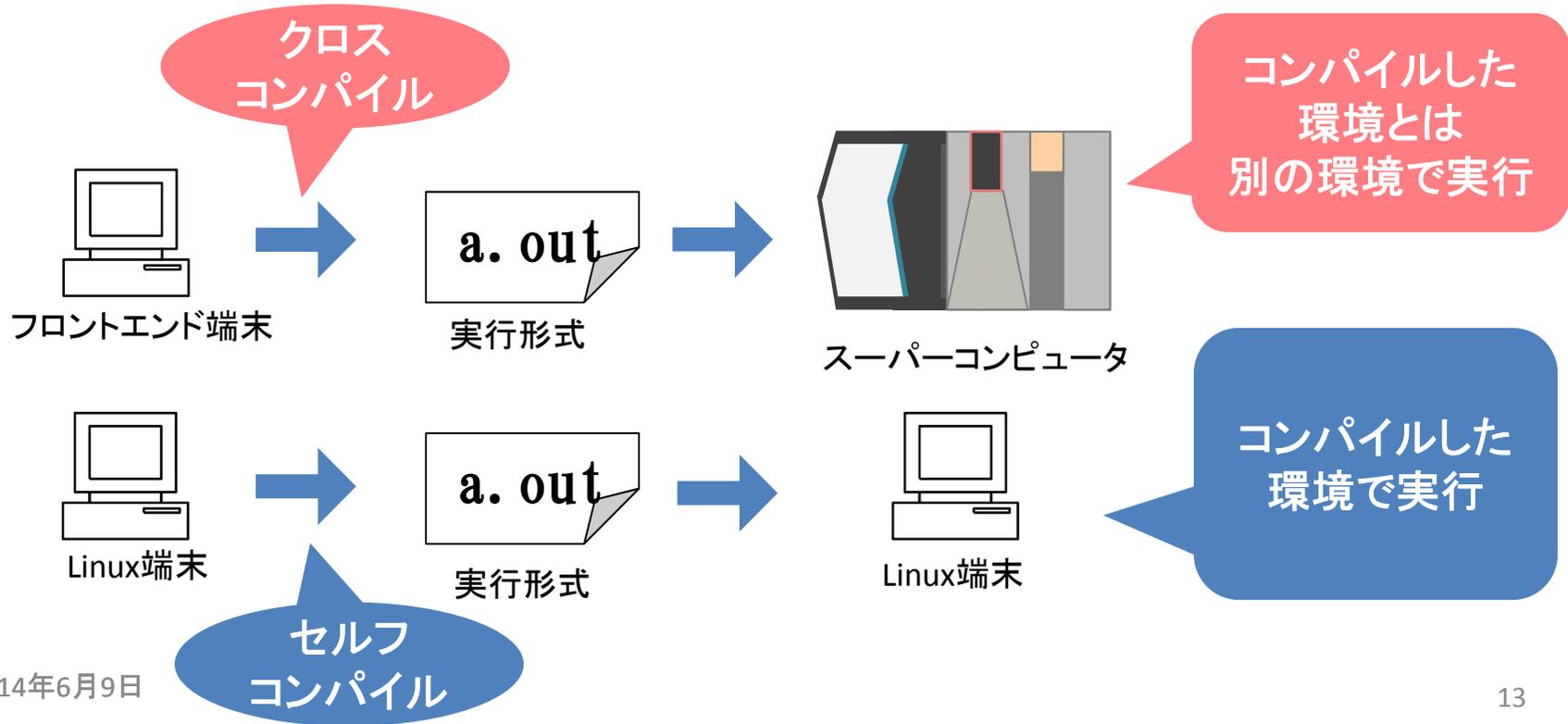
# Ⅲ コンパイル

# コンパイルとは

プログラムを機械語(計算機が実行できる形式)に変換すること

サイバーメディアセンターの計算機を利用する場合  
原則として「クロスコンパイル」を必要とします！

## クロスコンパイルとセルフコンパイル



# CMCのコンパイラ

## コンパイル作業

- SX, 汎用コンクラスタともフロントエンド端末でコンパイルする
- SX-9 はオプションが必要なので注意すること

## コンパイラの種類

	C	C++	FORTTRAN
SX用クロスコンパイラ	sxc++	sxc++	sxf90
	SX-9 用コンパイルは “-cfsx9” オプションを付けてください		
Intel コンパイラ (汎用コンクラスタ用)	icc	icpc	ifort

## SX用の実行形式であるかの確認方法

sxsymmap -f [実行ファイル]

## 実行に必要なとするメモリサイズの確認方法

sxsize [実行ファイル] (SX用クロスコンパイラ)

size [実行ファイル] (intel コンパイラ)

# SX用クロスコンパイラのオプション

- 翻訳リスト出力オプション：-Rn (FORTRANの場合)
  - n は 0~5 の数字で指定
    - -R0：リストを出力しない(規定値)
    - -R1：変形リストを出力する
    - -R2：変形リストと編集リストを出力する
    - -R5：編集リストを出力する
  - C/C++ コンパイラではオプションが異なる
- 簡易性能解析機能：-ftrace
  - 結果表示方法
    - ジョブスクリプト内で " setenv F\_FTRACE YES " を指定
    - プログラム実行後に sxftrace コマンドを実行 (解析情報ファイル(fttrace.out)が作成される)

参考

# SX用クロスコンパイラのオプション

- 並列化オプション：-P サブオプション
  - 並列化処理を使用する場合に指定
  - サブオプション: auto、openmp、multi 等
- 最適化オプション：-C サブオプション
  - ベクトル化、最適化のレベル指定
  - サブオプション: hopt、**vopt**、vsafe、ssafe、debug

レベル		説明		備考
		最適化レベル	ベクトル化レベル	
高	hopt	最大限	最大限	副作用の可能性あり
↑	vopt	最大限	既定レベル	既定値 副作用の可能性あり
	sopt	最大限	行わない	副作用の可能性あり(最適化)
↓	vsafe	行う(副作用を伴う可能性のある箇所以外)	行う(副作用を伴う可能性のある箇所以外)	
低	ssafe	行う(副作用を伴う可能性のある箇所以外)	行わない	
-	debug	行わない	行わない	デバッグ用

並列化や最適化についての詳しい解説は  
下記の講習会にてご説明します。

6月17日(火) 10:00~16:00  
「スーパーコンピュータと  
並列コンピュータの高速化技法の基礎」

講師は(株)日本電気が担当します。  
ぜひお申込みください！

## 演習2(コンパイル)

1. 下記演習プログラムを自分のホームにコピーしてください

ファイルパス: /sc/cmc/apl/kousyu/20140609/sample.f

例: % cp /sc/cmc/apl/kousyu/20140609/sample.f ~/

2. 「sample.f」をsx8用, sx9用にコンパイルしてください

例: % sxf90 -o sample-sx8.out sample.f

% sxf90 -cfsx9 -o sample-sx9.out sample.f

※「-o」を付けることで実行形式のファイル名を指定  
できます。指定しない場合、「a.out」が出力されます。

3. sx8 , sx9の実行形式とメモリサイズを確認してください

例: % sxsymmap -f sample-sx8.out

% sxsize sample-sx8.out

# IV ジョブスクリプト

# 利用形態（会話型とバッチ型）

## 会話型（インタラクティブ利用）

- コマンド等を通してコンピュータに直接命令し、リアルタイムで処理を実行する方法
- 操作として手軽

## 一括処理型（スケジューラ利用）

- コンピュータにまとめて処理を命令し実行する方法
- 処理の命令が終われば、ログアウトしてもOK

# 会話型

- SX-8Rのみ利用可能(利用方法はHP参照)
  - SX用プログラムの実行にのみ利用可
  - コンパイルやデータ圧縮は不可
    - スパコンはベクトル演算に特化しているので、これらの処理を苦手とします
    - フロントエンド端末をご利用ください

## 実行例

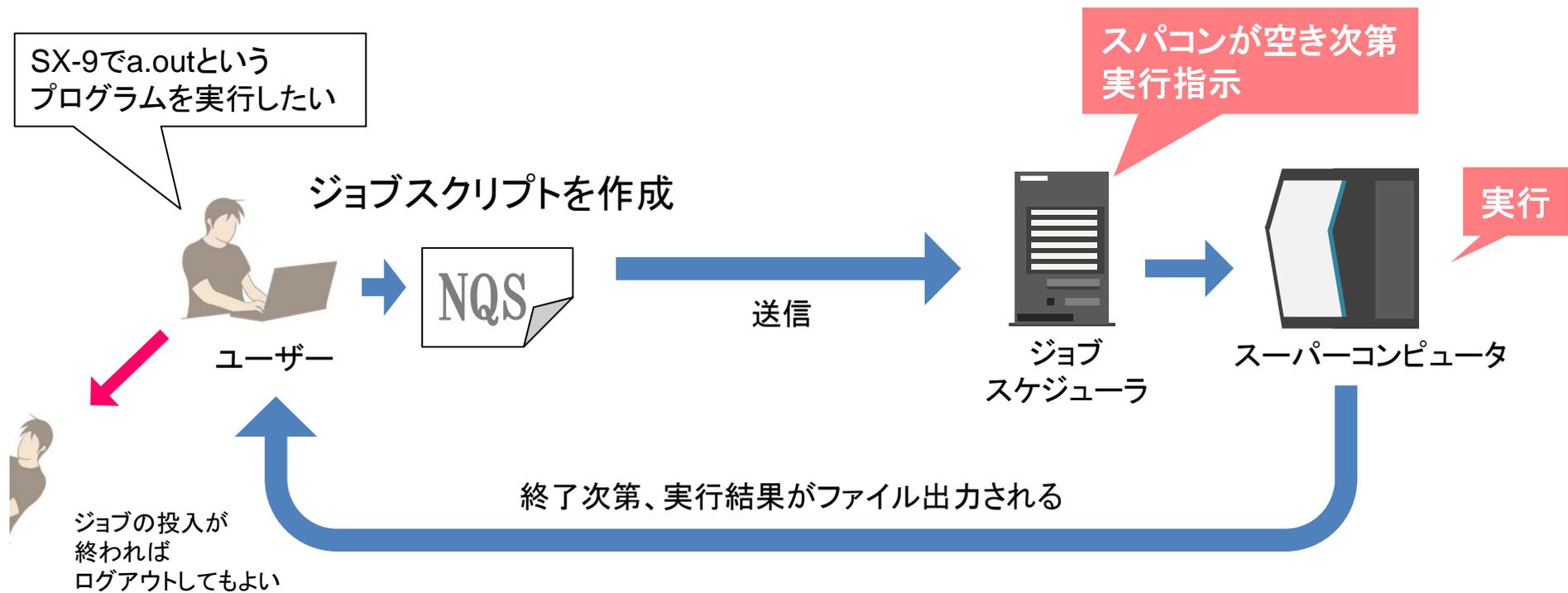
```
sx8% f90 sample.f  
f90: sample.f, _MAIN: There is 1 diagnosis.  
sx8% a.out  
2.000000 2.000000 2.000000 2.000000
```

~~コンパイル~~

実行&結果確認

# 一括処理型

- 基本的にはこちらを使う！
- コンピュータに実行させたい処理を「ジョブスクリプト」に記述
- ジョブスクリプトの内容に基づきスパコンが実行する



# ジョブスクリプト

## ジョブスクリプトの構成

- リソースや環境設定:「#PBS」で始まる
- スパコンに実行させる処理の記述:シェルスクリプト

## ジョブスクリプトの例

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

リソース、環境設定の指定

```
#PBS -q SX8F
```

```
#PBS -l cpunum_job=4, elapstim_req=5:30:00, memsz_job=10GB
```

```
cd $PBS_O_WORKDIR
```

```
./a.out > result.txt
```

スパコンに実行させる処理の記述

# リソース・環境設定

- 「#PBS」の後に記述する主なオプション

オプション	説明
-q	投入するジョブクラスの指定
-l	使用する資源値
	cpunum_job : 1ノードあたりのCPU数
	memsz_job : 1ノードあたりのメモリ量
	elapstim_req : ジョブの経過時間
	cputim_job : 1ノードあたりのCPU時間 (DBG, DBG9 のみ利用)
-o	標準出力の出力先ファイルの指定 (未指定時は [ジョブスクリプト名].o[リクエストID] )
-e	標準エラー出力の出力先ファイルの指定 (未指定時は [ジョブスクリプト名].e[リクエストID] )
-m	ジョブの状態の変更についてメール通知
	a : ジョブが異常終了したとき
	b : ジョブが開始したとき
	e : ジョブが終了したとき
-M	メールの通知先アドレスを指定
-v	環境変数の指定
-T	MPI 実行時に指定
	mpisx : MPI/SX 利用時
	mpich : MPICH 利用時
	mwjob : IntelMPI 利用時
-b	使用するノード数

# CMCのジョブクラス

システム	ジョブクラス	経過時間		CPU数		主記憶 (単位:GB)	
		既定値	最大値	既定値	最大値	既定値	最大値
		(単位:分)	(単位:時間)				
SX-8R	DBG	1 (1分)	1 (10分)	1	4	1	16
	SX8F(SXF)	1	24	1	8	1	120
	SX8L(SXL)	1	120	1	32	1	1000
SX-9	DBG9	1 (1分)	1 (10分)	1	4	1	128
	SX9	1	24	1	64	1	4000

「#PBS -q」の後に  
実行したいシステムの  
ジョブクラスを指定する！

# CMCのジョブクラス

システム	ジョブクラス	経過時間		CPU数		主記憶 (単位:GB)	
		既定値	最大値	既定値	最大値	既定値	最大値
		(単位:分)	(単位:時間)				
汎用コン クラス (HCC)	HCC-T	1 (1分)	制限なし※	2	128	4	256
	HCC-S	1	制限なし※	2	64	4	128
						12	384
HCC-M	1	制限なし※	2	16	4	32	

※汎用コンクラスは経過時間を無制限にしていますが、運用により2週間毎に各ジョブクラスを1日サービス停止しますので、実質、経過時間は最大12~13日までとなります！

# スパコンに実行させる処理の記述

- ファイルやディレクトリの実行・操作を記述する
- 記述方法はシェルスクリプト
- 主なNQSII 用の環境変数
  - \$PBS\_O\_WORKDIR : ジョブ投入時のディレクトリが設定される

## 標準出力／標準エラー出力の容量制限

⇒ SX-8R、SX-9: 30MB 汎用コンクスタ: 10MB

⇒ これ以上出力したい場合はリダイレクション(>)を利用

処理の記述の最終行に改行を入れること！

⇒ 未入力の場合、その行のコマンドが実行されない

# ジョブスクリプト解説

ジョブクラスの指定

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

```
#PBS -q SX8F
```

CPU数、経過時間、メモリサイズの指定  
コマ後にスペースを入れないよう注意！

```
#PBS -l cpunum_job=4, elapstim_req=5:30:00, memsz_job=10GB
```

```
cd $PBS_O_WORKDIR
```

ジョブ投入時のディレクトリへ移動

```
./a.out > result.txt
```

a.outを実行し、結果をresult.txtに出力する  
(リダイレクション)

## 演習3(ジョブスクリプト作成)

1. 下記のサンプルジョブスクリプトを自分のホームにコピー  
ファイルパス: /sc/cmc/apl/kousyu/20140609/sample.nqs

```
% cp /sc/cmc/apl/kousyu/20140609/sample.nqs ~/
```

※配布資料に誤りがあります。正しくは“20140609”となります。

2. 「sample.nqs」を元に sx8 , sx9 のジョブスクリプトを作成

```
% cp sample.nqs sample-sx8.nqs
```

```
% cp sample.nqs sample-sx9.nqs
```

- ・ジョブクラスを sx8 は DBG、sx9 は DBG9に変更
  - ・エディタは vi , emacs 等を利用
- 使い方が分からない人は手元の資料を確認

# V ジョブ投入

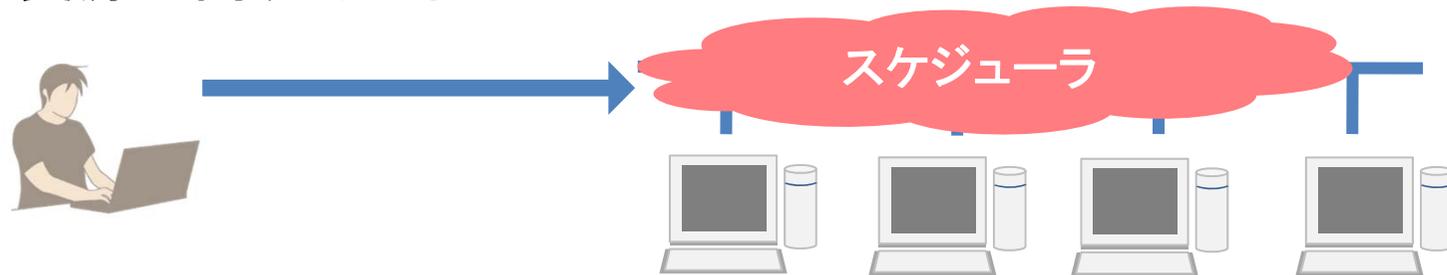
# ジョブ実行までの流れ

ジョブスケジューラはジョブスクリプトを受け、  
各スパコンにジョブの実行を指示する



# スケジューラとは

- **スケジューラ**: ユーザからのジョブ要求を受信し、あらかじめ管理者によって設定された資源割当ポリシーに従い、ユーザのジョブを計算資源に割り当てる



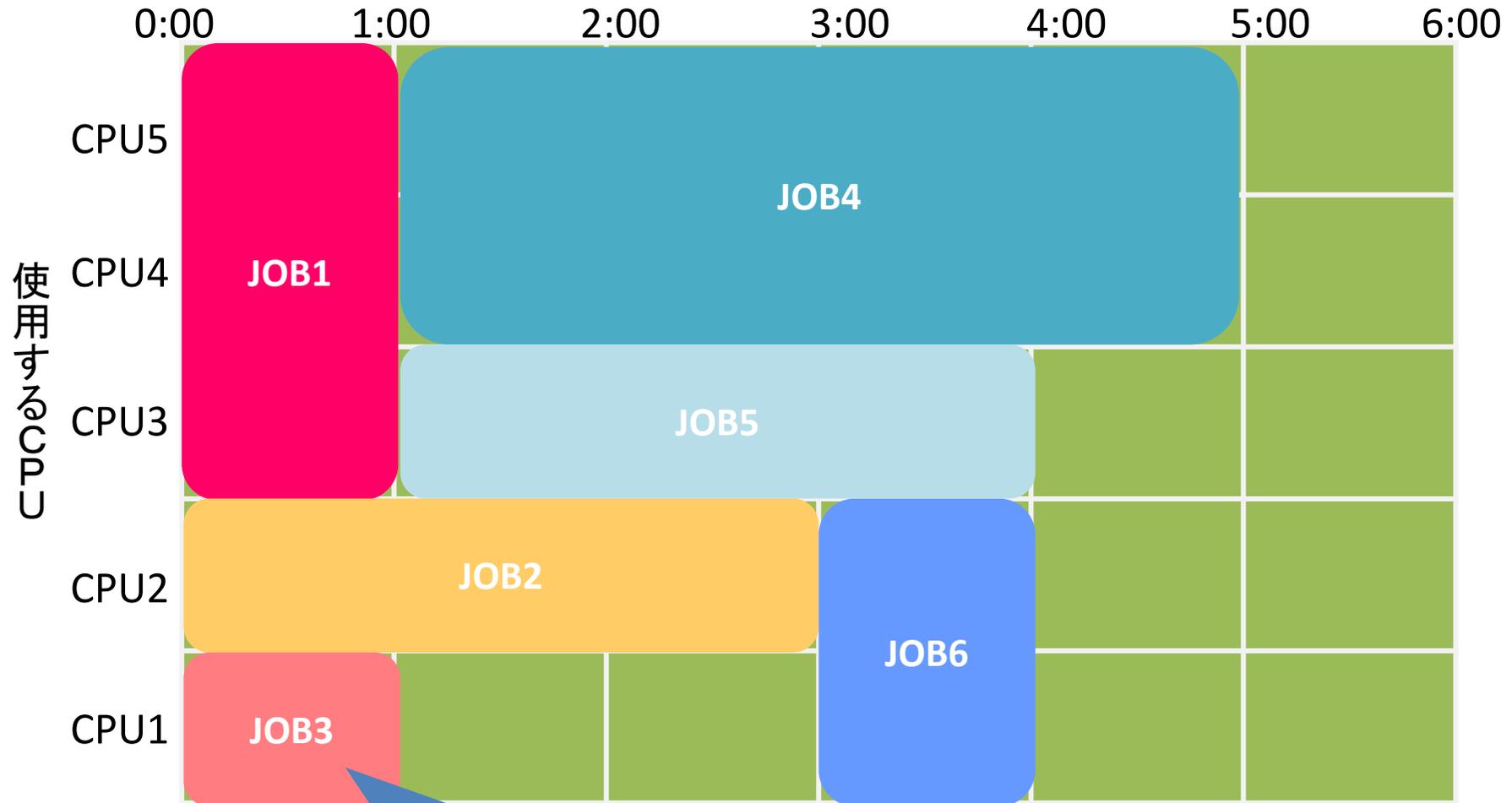
- クラスタを構成する計算機(ノード)の**静的情報**を把握
  - ディスク容量、メモリ容量、CPU性能、etc
- ノードの**動的情報**を定期的に監視、管理
  - CPU利用率、メモリ使用率、etc.
- ユーザより実行したいジョブ要求を受信
- ジョブを実行するのに適切なノードを選定
- ジョブ実行に伴う入力データのステージイン
- ジョブ実行後の出力データのステージアウト

# CMC のスケジューラ

- スケジュールの優先度の決定
  - シェア値：利用負担金コースごとに設定
  - 使用実績値：過去に利用したリソースの使用量
    - 経過時間、CPU数、メモリなど
  - ⇒ 優先度：各グループや個人のリソースの消費量がシェア値と比例するよう決定
- バックフィル型スケジューラ
  - ジョブの実行開始時間のマップを作成
  - マップに乗れば実行開始時間と経過時間が保障され、実行中は指定したリソースを占有して割り当てる

# スケジューラの例

ジョブが実行を開始する時間



優先度の高いジョブから  
スケジューリングされる

# ジョブの投入

- ジョブの投入
  - フロントエンド端末から投入
- 利用コマンド
  - % qsub [JobScriptファイル]
- 特殊な投入方法
  - リクエスト連携: 順番通りにジョブを実行したい場合に利用
    - % qsub [JobScript1] [JobScript2] ...
    - ※ 複数のジョブを同時に投入したい場合は注意

# ジョブの確認

- ジョブの予約状況の確認コマンド

- SX-8R : % sstat
- SX-9 : % sstat9
- 汎用コンクラスタ : % sstathcc

- 実行結果

RequestID	ReqName	UserName	Queue	Pri	STT	PlannedStartTime
12345.cmc	nqs-test	a61234	F4S	-1.1939	ASG	2013-03-03 01:23:45

## 状態監視

実行時刻が決まる(アサインされる)と「ASG」表示になる。  
混雑具合や優先度により、ジョブアサインまでの時間が異なるが、一旦アサインされるとその時刻にジョブ実行が始まる。

## 実行開始時刻

システムメンテナンスやトラブル時は、再スケジュールされることをご了承ください。

# ジョブの確認

- ジョブの状態確認コマンド  
`% qstat`

- 実行結果

```
fronta% qstat
```

RequestID	ReqName	UserName	Queue	STT	Memory	CPU	Elapse
12345.cmc	nqs-test	a61234	F4S	RUN	8.72G	830.66	208

## QUE(待ち状態)とRUN(実行中)

待ち状態では「QUE」  
実行が始まると「RUN」となる。

## CPUとElapse

CPUは、実際にジョブが消費した時間を指す。  
複数CPU指定の場合は、全CPUを累積表示する。  
Elapsは、ジョブが実行されてからの経過時間を指す。

# ジョブの確認

- ジョブのキャンセル

`% qdel [RequestID]`

- その他のコマンド

- `jobr`: 状態確認の際、`qstat`だと自分のジョブのみ表示されるが、`jobr`コマンドを使えば、他の利用者ジョブも含めた状態が確認できる。

- `jobstat`: 実行中及びスケジュール済みのジョブが一覧で表示され、両方の状況が一度に確認できる。

# 演習4(ジョブ投入)

1. 作成したジョブスクリプトを使用してジョブを投入

`% qsub sample-sx8.nqs`

`% qsub sample-sx9.nqs`

2. 投入したジョブの状態を確認

`% sstat` :sx8予約状況確認

`% sstat9` :sx9予約状況確認

`% qstat` :ジョブの状態確認

3. 結果ファイルの確認

- ・cat コマンドを使用

- ・SX については標準エラー出力ファイルの proginfo 等

利用についてのお問い合わせは  
大阪大学 情報推進部 情報基盤課  
研究系システム班  
system@cmc.osaka-u.ac.jp

までご連絡下さい！

## よくある質問

今までよくあった質問について説明をまとめてみました

読んでいただいて分からないことがあったらいつでもご質問ください  
(大歓迎です)

質問いただいたことは、テキストやWEBの説明に反映させたいと思っています。

**「分からない」は財産です**

出典:大阪大学レーザーエネルギー学研究中心 福田優子氏  
「初めてプログラムするための基礎知識」

# 1 計算機の内部表現

(SX:float0形式(IEEE形式)の場合)

1ビット



1と0で表現

1バイト



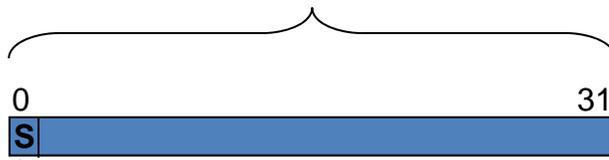
1word(8ビット) aやbなどの文字は1バイトで表される

※

Sは符号ビット  
0:正または0  
1:負

4バイト(32ビット)

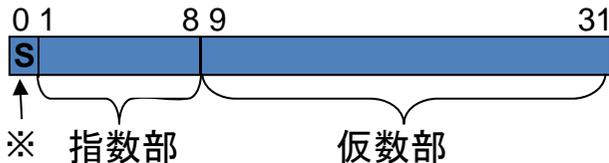
4バイトの  
整数  
(1)



**単精度**

有効数字 10進数 約7けた  
約 $10^{-38}$ から $10^{37}$

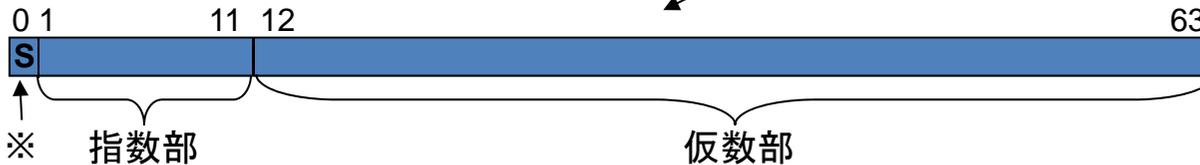
4バイトの  
実数  
(1.0e0)



**倍精度**

有効数字 10進数 約16けた  
約 $10^{-308}$ から $10^{308}$

8バイトの  
実数  
(1.0d0)



8バイト(64ビット)

## 2 書式付データと書式なしデータ(アスキーとバイナリ)

1と0で表現

### 書式つきデータ (アスキー、文字)

計算機の内部表現を変換して出力

```
write(n, * ) a,b  
write(n,100)a,b
```

```
write(6,100) a  
100 format(1x,e11.4)
```

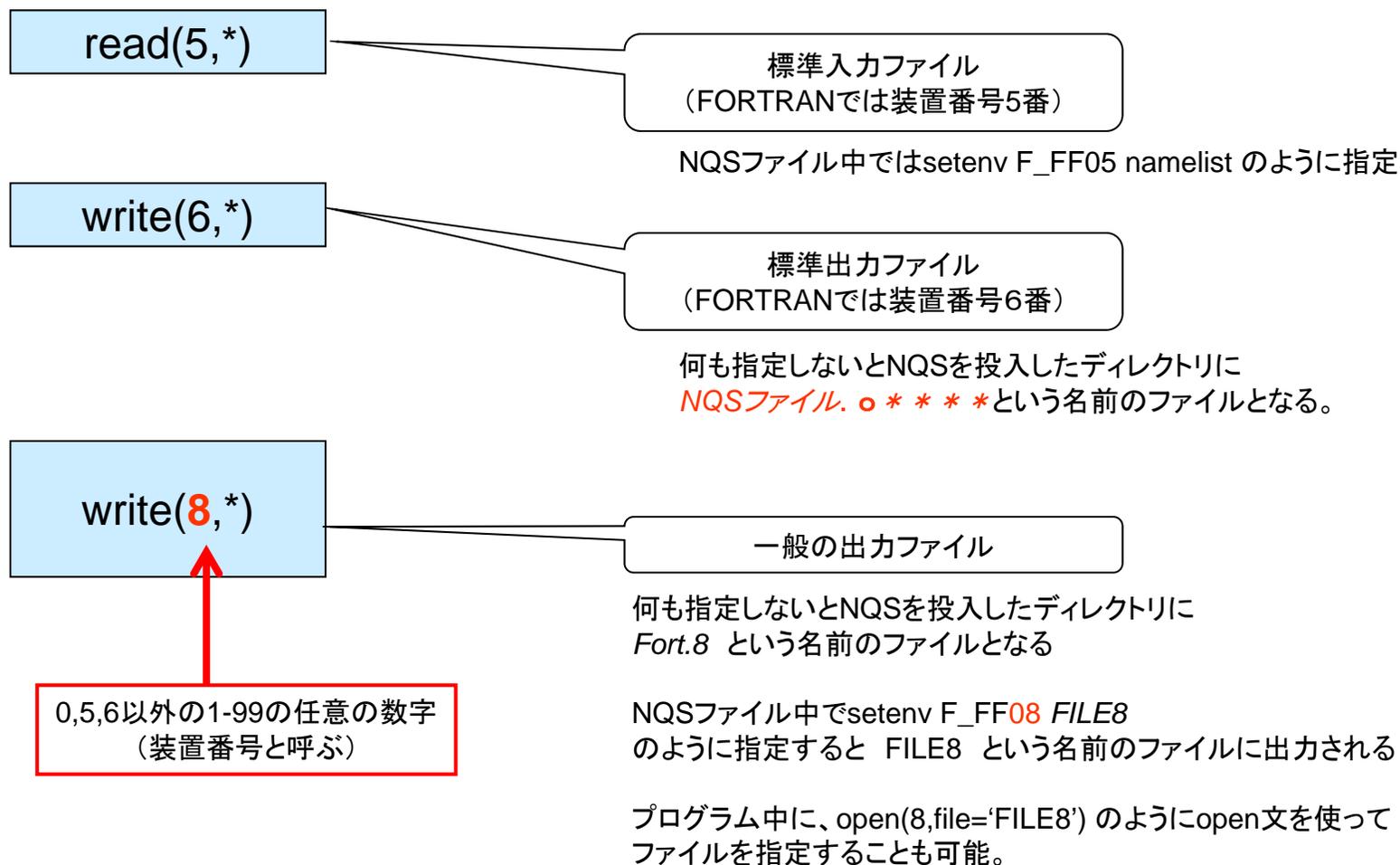
### 書式なしデータ (バイナリ)

計算機の内部表現のままのデータ

```
write(n) a,b
```

### 3 プログラム中のread & writeとファイルの関係

(FORTRANプログラムの入出力)参考:FORTRAN90/SXプログラミングの手引き



# FORTRAN入門(田口先生)のテキストお勧めです

摂南大学 理工学部 電気電子工学科 田口 俊弘

## 1) Fortranでシミュレーションをしよう

阪大レーザー研のホームページで公開中(71ページ)  
研究室むけのテキストを一般的に書き直して提供いただきました  
2011年6月初版

## 2) Fortranスマートプログラミング

東北大学サイバーサイエンスセンターの広報誌で連載開始  
より初心者むけに書き直して、2012年7月より  
3回に分けて連載完了(24ページ/一回目)

## 3) Fortranスマートプログラミング 2013

完結したものに加筆し、100ページを超えるテキストができ、レーザー研の  
ホームページで公開しました。  
勉強したくなったときに、最新バージョンをチェックしてみてください。

- ・本当にシミュレーションをバリバリされている先生の視点で実際のプログラミングで必要なことがすべて説明されている。(4章までが初心者向け、5章6章はなれてから)
- ・読みやすいプログラムの例が具体的に説明されている。スマートプログラムのノウハウが掲載されている
- ・付録もよい  
初心者が知りたいことが書かれている  
(gfortranを用いたコンパイルから実行まで、エラー・バグへの対処法など)

### 1.3.2 メモリ上での配列の並び

配列はコンピュータ内部において連続したメモリ領域で実現されています。例えば、`real a(10)`と宣言された配列は、図 1.2 左のように、`a(1)`、`a(2)`、...、`a(10)`の順で並んで実数型のメモリです

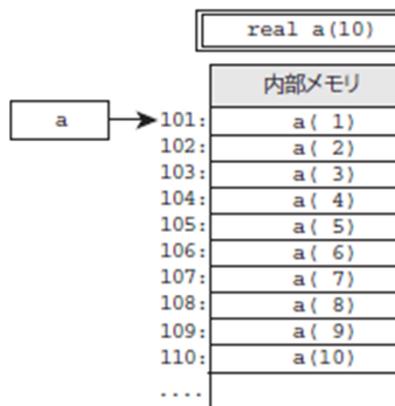


図 1.2

### 1.4 プログラミング スマートテクニック (その1)

計算手順が複雑になったり大量のデータを処理するようになると、数式を単純にプログラムに置きかえるのではなく、計算機の特徴を考慮したプログラムにすることで計算効率や精度を高めることができます。ここでは、計算速度を向上させるテクニックや、コメントを入れてプログラムのメンテナンスを容易にする手法などについて説明します。

#### 1.4.1 コンピュータで表現可能な数値の大きさ

我々が日常的に使っている 10 進数は、10 のべき乗が基本です。すなわち、数字を 10 の多項式で表したときの係数を次数の大きい方から並べたものです。ただし、係数は 10 より小さい整数 (0~9) でなければなりません。例えば、1203 という数は、10 の多項式で表せば、

$$1203 = 1 \times 10^3 + 2 \times 10^2 + 0 \times 10^1 + 3 \times 10^0$$

となるので、表現が“1203”なのです。

これに対し、コンピュータ内部で使われている 2 進数は、2 のべき乗を基本として表された数です。この場合、多項式の係数は 2 より小さい整数でなければならないので、0 か 1 です。例えば、10 進数で 123 と表示される数を 2 の多項式で表せば、

#### 1.4.4 $\pi$ の作り方

物理や数学の計算をするときには円周率 $\pi$ を使うことが多いです。 $\pi$ の値を倍精度計算用に有効数字16桁で表せば、

```
real pi
pi = 3.141592653589793
```

となります。これをそのまま使えば良いのですが、この面倒です。そこで逆三角関数を使って計算

```
real pi
pi = acos(-1.0)
pi = 2.0*asin(1.0)
pi = 4.0*atan(1.0)
```

などのように書くことができます。関数計算にはするのは1回だけです。問題になるほどではありません。

#### 1.4.5 コメント文の挿入

Fortranでは“!”の後に続く文字は全て無視されます。すなわち何を書いても実行とは無関係です。これをコメント文といいます。コメント文を機会あるごとにプログラム中に入れて、書かれている内容を表示するように心がけましょう。さもなくば、プログラムが長くなるにつれて、自分でも何を書いたのか忘れてしまいます。例えば、

```
! area of circles
s = pi*r*r
```

のように、1列目に“!”を書けば、その行はコメント行になります。また、

```
s = pi*r*r      ! 円の面積
v = 3*pi*r*r/4  ! 球の体積
```

のように、実行文の末尾に書き込むこともできるし、日本語を書くこともできます。ただし、日本語環境によっては正しく認識できずに文字化けすることがあります。可読性を考慮するならローマ字つづりでも良いから半角英数字だけでコメントを書いた方が無難です。

### 3.1 サブルーチンの利用目的

“ルーチン”とは、動作開始点と終了点が定義されている閉じたプログラム手順を示す用語です。プログラムを起動したときに最初に動作するのはメインプログラム

です。“サブルーチン”は、メインプログラムと同様にプログラムと異なり、単独で動作することはできません。必要に応じて実行させる必要があります。一つのプログラムは名前が異なれば複数存在してもかまいません。サブルーチンを作成して別々にコンパイルし、必要に応じて結合(リンク)して実行します。サブルーチンを利用する目的は主として3つあります。

- (1) 複数の場所で同じ計算手順を使用するため
- (2) 方程式の解法や行列式の計算などのような定型処理をまとめるため
- (3) 長いプログラムを分割してメンテナンスを容易にするため

上記のプログラム実行の流れを図3.1に示します。

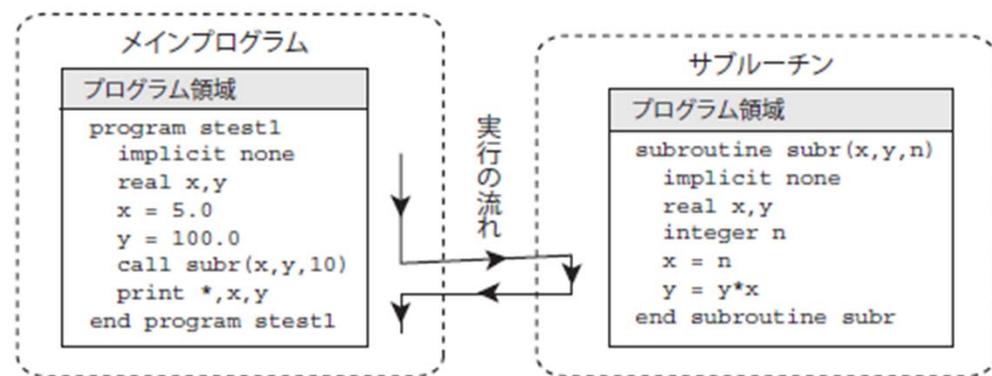


図 3.1 サブルーチンの呼び出しと実行の流れ

CMP

大阪大学レーザーエネルギー学研究中心 高性能計算機室のページです

---

### 公開テキスト

<a href="#">高性能計算機室の紹介</a>	<ul style="list-style-type: none"> <li>■ <a href="#">- 初心者向け - 「メールとネットワークの基礎」</a></li> <li>■ <a href="#">シミュレーション (はじめて科学技術計算をされる方へ)</a></li> <li>■ <a href="#">並列化のいろはを勉強したい方へ HPF講習会テキスト</a></li> <li>■ <a href="#">高性能計算機室作成テキスト一覧(非公開)</a></li> </ul>
<a href="#">スパコンを用いた研究の紹介</a>	
<a href="#">講習会のお知らせ</a>	
<a href="#">ILE計算機システム</a>	<ul style="list-style-type: none"> <li>■ <a href="#">- 初心者向け - 「メールとネットワークの基礎」</a></li> </ul>
<a href="#">CMCスパコンシステム情報</a>	
<a href="#">共同研究者用ネットワークについて</a>	
<a href="#">公開テキスト</a>	
<a href="#">スーパーコンピュータシンポジウム</a>	
<a href="#">CMPトップページ</a>	



- [シミュレーション \(はじめて科学技術計算をされる方へ\)](#)

このテキストは、大学や大学院で初めてパーソナルコンピュータや大型の計算機などを用いて計算しようという方を対象に、基礎知識を説明したものであるという方も、スーパーコンピュータで培われた、これは気をつけましょうという知識を持って利用していただきたいと専門家の方々に相談したところ、下記でご紹介させていただきますので、も

また、自分でプログラムを書いて研究されている方は、FORTRANを利用されていることが多いですが、FORTRANのよい入門書がないと相談したところである摂南大学の田口先生が、ご自分の研究室向けに作られていた入門書を一般的なものに加筆修正して提供してくださいました。質問やコメント性能計算機室(e-mail:ile-comp@attマークile.osaka-u.ac.jp)までお寄せ下さい。

発行	Version	初版年・製作者	公開テキスト
2011/10/17	第2.1版	2011/7/8(福田)	<a href="#">パソコン&amp;スーパーコンピュータで計算するための基礎知識</a> pdf 1.17MB
2013/5/20	第3版	2013/5/20 (摂南大学 田口俊弘先生より ご提供いただきました)	<a href="#">Fortranスマートプログラミング(2013年度版)</a> pdf 892KB

発行





## 東北大学サイバーサイエンスセンター 大規模科学計算システム

サイバーサイエンスセンターは  
教育研究機関・民間研究機関のための  
高性能コンピューティング基盤を提供しています

本館案内・お問合せ | [FAQ](#) | [リンク](#) | [サイバーサイエンスセンター](#) | 東北大学

### センターからのお知らせ

[過去のお知らせ](#)

- 2012-09-12 利用負担金額の表示コマンドについて
- 2012-09-12 負担金の支払い費目について
- 2012-09-12 計算科学・計算機科学人材育成のためのスーパーコンピュータ無償提供制度について(再掲載)
- 2012-08-29 非線形構造解析汎用プログラム MSC.Marc2012/Mentat2012のサービス開始について
- 2012-08-28 コンパイラアップデートのお知らせ
- 2012-08-20 ユーザの研究紹介掲載のお知らせ  
大規模科学計算システムニュース (最新号: No.140, 2012.9.14 発行)

- 2012-08-03 計算機システムのサービス休止について
- 2012-08-01 反応経路自動探索プログラム GRRM110のサービス開始について
- 2012-08-01 平成24年度講習会案内(9月開催分)
- 2012-08-01 夏季中の利用相関について
- 2012-08-01 夏季休業に伴う休館について
- 2012-07-17 平成24年度講習会案内(8月9月開催分)

[メールマガジンの登録](#)

### スーパーコンピュータ SX-9



### 並列コンピュータ Express5800



### アプリケーションサービス



### 共同研究



### 利用案内・申請方法

#### 学術利用の方

#### 民間企業利用の方

#### 各種申請用紙



### 講習会案内



### 利用相談



### 成果報告



### センター施設利用



# 利用の手引きなど

- 大規模計算機システムの簡単な利用方法  
[http://www.hpc.cmc.osaka-u.ac.jp/j/service/front\\_guide.html](http://www.hpc.cmc.osaka-u.ac.jp/j/service/front_guide.html)
- 利用負担金表  
<http://www.hpc.cmc.osaka-u.ac.jp/j/futankin/index.html>
- ジョブクラス表  
<http://www.hpc.cmc.osaka-u.ac.jp/j/service/jobclass.html>
- ジョブの予約状況  
[http://www.hpc.cmc.osaka-u.ac.jp/j/service/sx\\_jobjyoho.html](http://www.hpc.cmc.osaka-u.ac.jp/j/service/sx_jobjyoho.html)

# Linux環境からの接続

- フロントエンド端末へのログイン

- Linux 環境から接続する場合

1. コンソールを開く

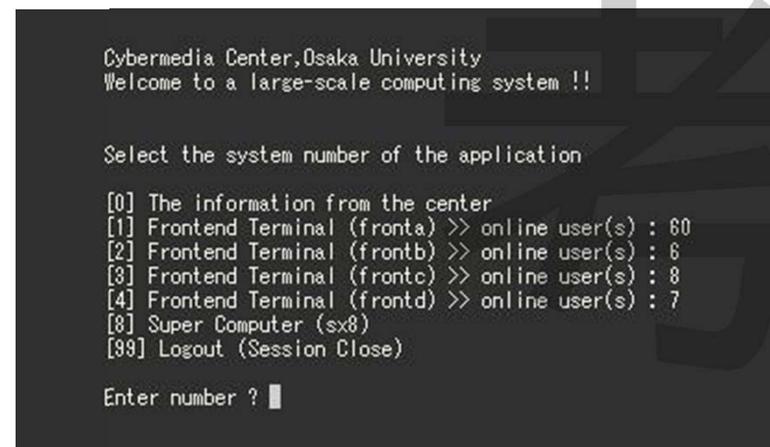
2. 下記のコマンドを入力して実行

**ssh username@login.hpc.cmc.osaka-u.ac.jp**

↑ 自分のログインIDに置き換えてください

3. **パスワード**を入力

4. メニュー画面が表示されるので  
[1]～[4] のどれかを選択



```
Cybermedia Center,Osaka University
Welcome to a large-scale computing system !!

Select the system number of the application

[0] The information from the center
[1] Frontend Terminal (fronta) >> online user(s) : 80
[2] Frontend Terminal (frontb) >> online user(s) : 6
[3] Frontend Terminal (frontc) >> online user(s) : 8
[4] Frontend Terminal (frontd) >> online user(s) : 7
[8] Super Computer (sx8)
[99] Logout (Session Close)

Enter number ? █
```

# ファイル転送

- ファイル転送

- WinSCP を利用した場合

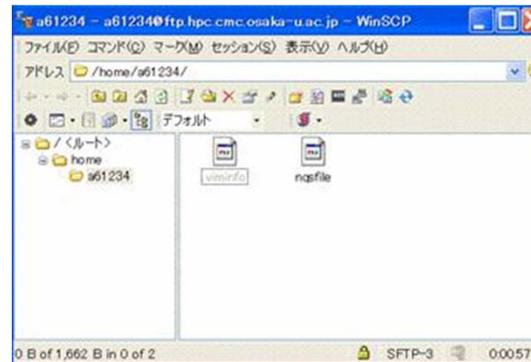
- 1. ホスト名に **ftp.hpc.cmc.osaka-u.ac.jp** を入力

- 2. **ユーザ名とパスワード**を入力

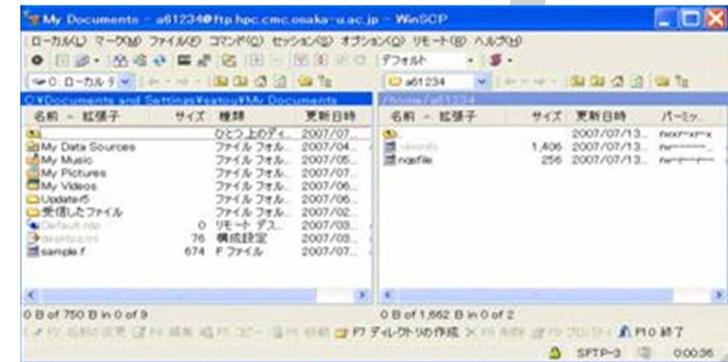
参



ログイン画面



ログイン後  
(ノートンタイプ)



ログイン後  
(エクスプローラタイプ)

# 各種サポート

利用方法について疑問が生じたら・・・

- ・ 本センターホームページで公開している利用の手引き・マニュアルを閲覧  
※ マニュアルはユーザ認証があるので、利用者番号・パスワード が必要です。



それでも判らない、期待した内容が掲載されていない場合

- ・ システム管理者やプログラム相談員への問い合わせ  
hpc-support @ hpc.cmc.osaka-u.ac.jp 宛にメールで質問してください。  
※ 問い合わせメールには 利用者番号 を記入ください。

計算機が止まっていると感じたら・・・

- ・ システムメンテナンスや停電によるサービスの停止では？  
停止直前まで、ホームページおよびログインサーバへの接続時の案内でお知らせしております。
- ・ 通信障害では？  
研究室(利用者側)からの通信を確認してください。メールの送受信はできるか？ 他のHPは見られるか？  
これらができない場合は、本センター以外での障害が予測されます。
- ・ 該当しない場合は？  
システム管理( system @ cmc.osaka-u.ac.jp ) までご連絡ください。

# 使用上の注意

## 文字コードについて

スーパーコンピュータでは、EUC 文字コードを使用しています。パソコン(PC)では、SJIS コードが使用されているため、プログラムやデータのアップロード後はコード変換が必要です。

フロントエンド端末にて、SJISコードのファイル(sjisfile) を EUCコード(eucfile)に変更する

```
fronta% nkf -eLu sjisfile > eucfile
```

(なお、この作業はファイル転送ソフトの設定によっては、不要となる場合もあります。)

## ジョブの投入方法について

- ・ 使用リソース(CPU数やメモリーサイズ)に見合ったジョブクラスへジョブ投入する。

1CPUしか使わないジョブを、複数CPU利用のクラスに投入すると、システム的には多くのCPUを確保しようとするため、全体的な効率が悪化し、他の利用者に迷惑が掛かります。また、メモリー容量も大きく宣言すると、システムに空き領域ができるまでジョブがスタートできません。実容量を確認し、投入すること。



これらの事はスーパーコンピュータのような他の利用者が共有して使うシステムにおいては、マナーとしてご理解ください。

これに加え、JM(JobManipulator)では、ジョブ毎のElapsTime(経過時間)を用いてバックフィル型のスケジュールをするため、ElapsTime もできる限り正確に制限してください。

# Web利用申請システム

下記のページから新規利用申請・パスワード初期化が可能です。

<https://portal.hpc.cmc.osaka-u.ac.jp/ouweb>

大阪大学  
OSAKA UNIVERSITY

Web利用申請システム

利用申請のご案内

利用申請をする      パスワード初期化申請をする      終了する

**【利用申請】**

1. 利用申請(登録)画面で、申請に必要な項目を入力して利用申請を行って下さい。
2. 利用申請を行うと、申請者とその関係者、およびセンター管理者へ申請が行われたことをメールで通知します。
3. メールで、申請書を印刷するためのワントタイムURLが通知されます。**30日**以上経過すると無効になります。
4. 利用申請に対し、センター管理者により「承諾」、「修正依頼」の処理がされます。この結果が、申請者にメールで通知されます。
5. 修正依頼があった場合は、通知メールに記載されたワントタイムURLより、申請内容の修正が可能となります。必要な修正を行い、再度申請して下さい。
6. 修正のためのワントタイムURLは**24時間**以上経過すると無効になります。
7. 申請が承諾された場合は、通知メールに記載されたワントタイムURLより、パスワード初期化を行って下さい。これにより『全国共同利用ポータル』へのログインが可能となります。以後のご利用は、『全国共同利用ポータル』より行って下さい。
8. このパスワード初期化のワントタイムURLは、**5日**以上経過すると無効になります。
9. 一旦、利用申請を行ったが、申請を取り下げたい場合は、センター管理者へ直接連絡して下さい。

**【パスワード初期化申請】**

1. パスワードを失念した場合は、パスワード初期化申請画面より、初期化を申請して下さい。
2. パスワード初期化申請を行うと、申請者とセンター管理者へ申請が行われたことをメールで通知します。
3. 申請者への通知メールに記載されたワントタイムURLより、パスワード初期化を行って下さい。
4. パスワード初期化のワントタイムURLは、**30分**以上経過すると無効になります。

Copyright © 2012 OSAKA UNIVERSITY. All Rights Reserved.