

# Advanced language features

- Macros and tricks for your instrument...

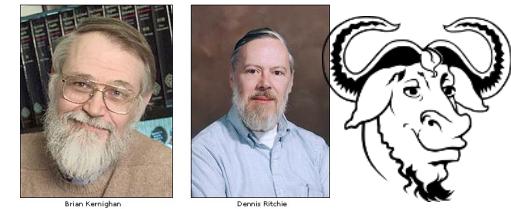


# DECLARE / INITIALIZE

- Use the DECLARE section define user variables and functions.
  - `DECLARE %{`
  - `double some_global_var;`
  - `%}`
- Use USERVARS for particle-dependent flags (**McStas v3 specific, use DECLARE in v2**)
  - `USERVARS %{`
  - `double myvar;`
  - `%}`
- Use INITIALIZE for initialization of user variables and calculations.
  - `INITIALIZE %{`
  - `myvar = sqrt(PI*input_var)*rand01();`
  - `%}`
- - Both use normal c-syntax.
- BEWARE: (example) What you do in the c-style areas is c-standard, e.g. trigonometric functions from `math.h` use radians! - McStas placement specifiers work in degrees, etc...



K & R. / GNU



SEGUNDA EDICION  
EL  
 LENGUAJE DE  
PROGRAMACION  
Con base  
en el ANSI C

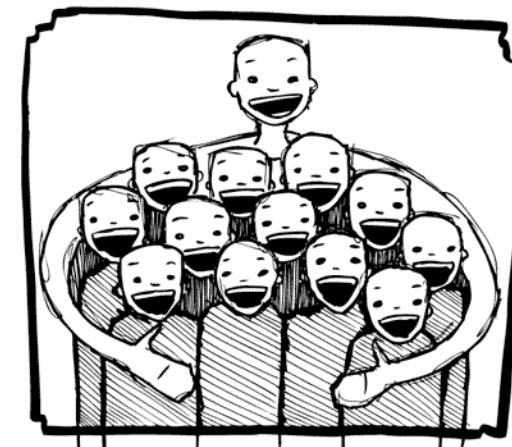


BRIAN W. KERNIGHAN  
DENNIS M. RITCHIE



# %include

- Instrumentfiles can include external c-code or other instrumentfiles... See the examples:
- ILL\_H15\_IN6.instr:%include "monitor\_nd-lib"
- ILL\_H16\_IN5.instr:%include "ILL\_H16.instr"
- ILL\_H25\_IN22.instr:%include "ILL\_H25.instr"
- ILL\_H25\_IN22.instr:%include "templateTAS.instr"
- Used in the DECLARE section



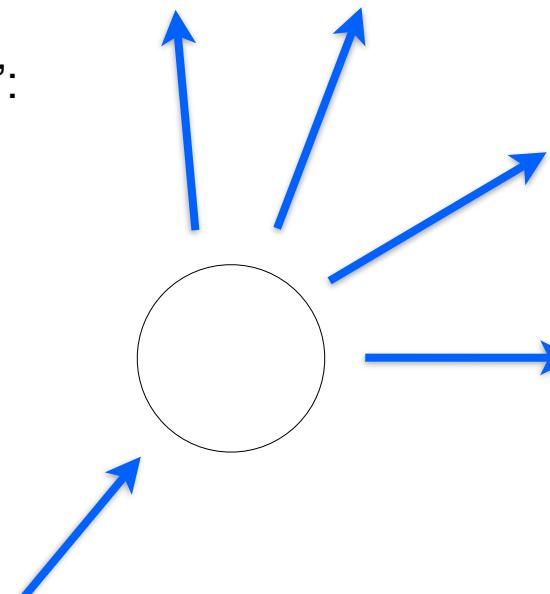
# Syntax in one, complex view...

```
{SPLIT} COMPONENT name = comp(parameters) {WHEN condition}  
AT (...) [RELATIVE [reference|PREVIOUS] | ABSOLUTE]  
{ROTATED {RELATIVE [reference|PREVIOUS] | ABSOLUTE} }  
{GROUP group_name}  
{EXTEND C_code}  
{JUMP [reference|PREVIOUS|MYSELF|NEXT] [ITERATE number_of_times | WHEN condition] }
```



# SPLIT

- Increase statistics beyond this point in the instrumentfile
- SPLIT n MyArm = Arm()
- AT somewhere
- will “formulate an if-statement”:
  - for j=1:n
  - comp1
  - comp2
  - comp3
  - ...
  - end (of instrument)
- ONLY meaningful in case of Monte Carlo choices after SPLIT point...



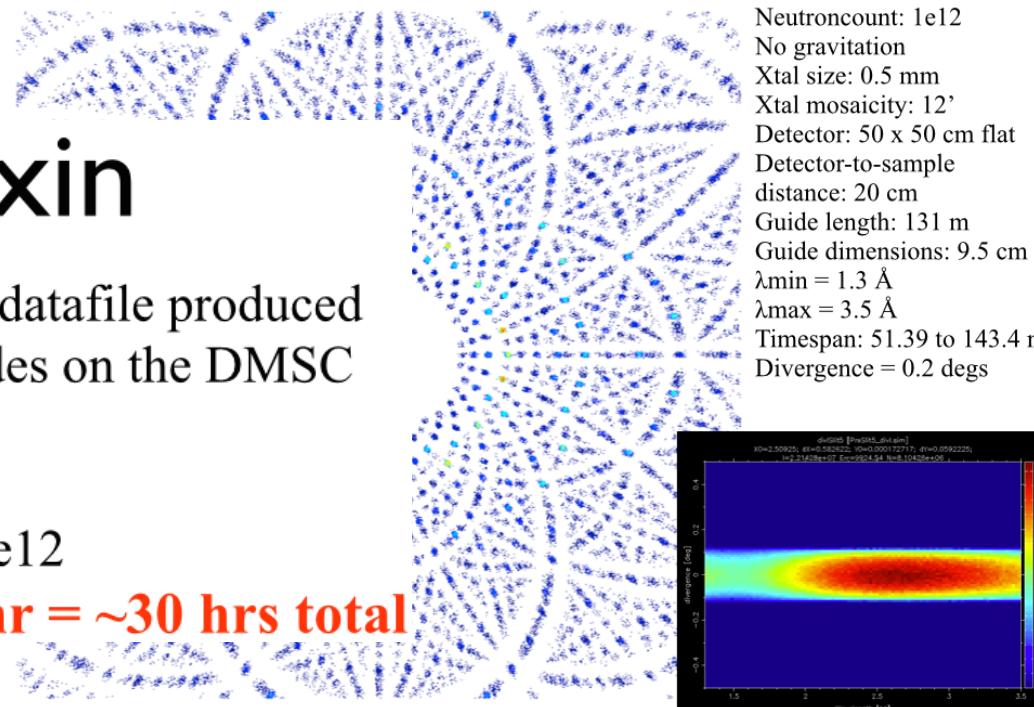
# Problem: McStas Single\_crystal.comp “slow” for large unit cell diffraction studies

- Example: Rubredoxin

1 timebin, 1000 x,y-bins

## Rubredoxin

Images created from simulated datafile produced August 20th 2012 using 25 nodes on the DMSC cluster.

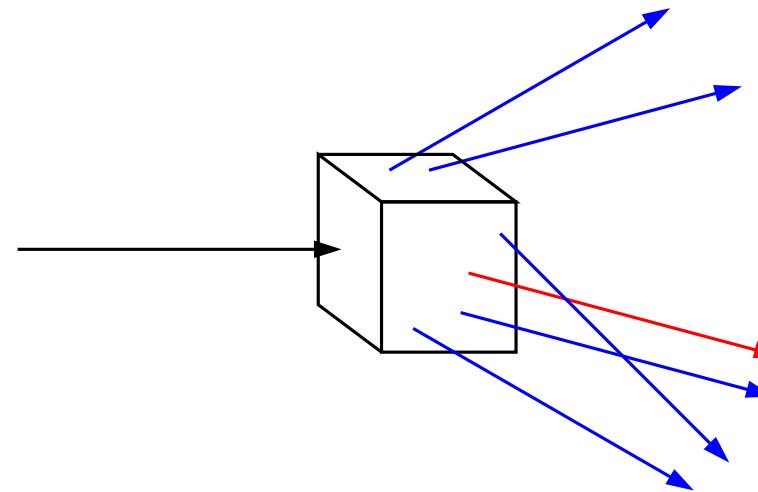


Neutron count: 1e12  
**Simulation time: ~10 + ~20 hr = ~30 hrs total**

- Reflection list ~ 124 K reflections (still “small” in the PX world!!)

# Algorithm improvement: Use incoming neutrons more efficiently - scatter each one on all possible reflections

- **Red:** Original algorithm, one incoming neutron used only once
- **Blue:** Improved algorithm, each incoming neutron scattered (via SPLIT keyword) all possible times
- Component makes **estimate on average number of “active” diffraction spots** - in the case Rubredoxin this is around **50!**



!!! For now, does not work in our GPU implementation !!!

# GROUP - components working in parallel



COMPONENT Mono1 = Monochromator\_curved(...)

*AT (0,0,-LMM) RELATIVE Cradle ROTATED (0,A1/2,0) RELATIVE Cradle*  
*GROUP IN6Monoks*

COMPONENT Mono2 = Monochromator\_curved(...)

*AT (0,0,0) RELATIVE Cradle ROTATED (0,A2/2,0) RELATIVE Cradle*  
*GROUP IN6Monoks*

*- One comp after the other is “tried” in sequential order until the neutron was SCATTERED.*

# EXTEND



- Enrich component behaviour using EXTEND:

```
COMPONENT Mono1 = Monochromator_curved(...)
AT (0,0, -LMM) RELATIVE Cradle ROTATED (0,A1/2,0) RELATIVE Cradle
GROUP IN6Monoks
```

```
EXTEND
```

```
%{
    if (SCATTERED) { myvar = 1; }
%}
```

```
...
```

```
COMPONENT Mono2 = Monochromator_curved(...)
AT (0,0, 0) RELATIVE Cradle ROTATED (0,A2/2,0) RELATIVE Cradle
GROUP IN6Monoks
```

```
%{
    if (SCATTERED) { myvar = 2 ;}
%}
```

K & R. / GNU



SEGUNDA EDICION  
EL  
 LENGUAJE DE  
 PROGRAMACION  
Con base  
en el ANSI C

C

BRIAN W. KERNIGHAN  
DENNIS M. RITCHIE

# WHEN

- Syntax:

```
COMPONENT Mine = Yours(blah, blah)
WHEN (c-expression) AT (....)
```

- Is very powerful when combined with EXTEND and user variables, or as a method to let input parameters select if certain components are active.

K & R. / GNU

- Example: Use EXTEND to flag if neutron was scattered on one monochromator or another. Then later use WHEN to only show contribution from blade N at sample



```
COMPONENT Mon = PSD_monitor(...)
WHEN (myvar==1) AT (0,0,0) RELATIVE Sample
```

SEGUNDA EDICIÓN  
EL  
 LENGUAJE DE  
 PROGRAMACIÓN  
Con base  
en el ANSI C

C

BRIAN W. KERNIGHAN  
DENNIS M. RITCHIE

# JUMP

- A goto. Be careful. Can be used in two situations:
- JUMP to myself
- JUMP to an Arm
- No coordinate transformations are applied... (Meaning that if the Arms you JUMP between do not coincide you will “move” / “reorient” the neutrons...)
- Syntaxes:
- COMPONENT a=b(...)
- WHEN (expr) AT (...) JUMP somewhere
- COMPONENT a=b(...)
- WHEN (expr) AT (...) JUMP myself



# JUMP

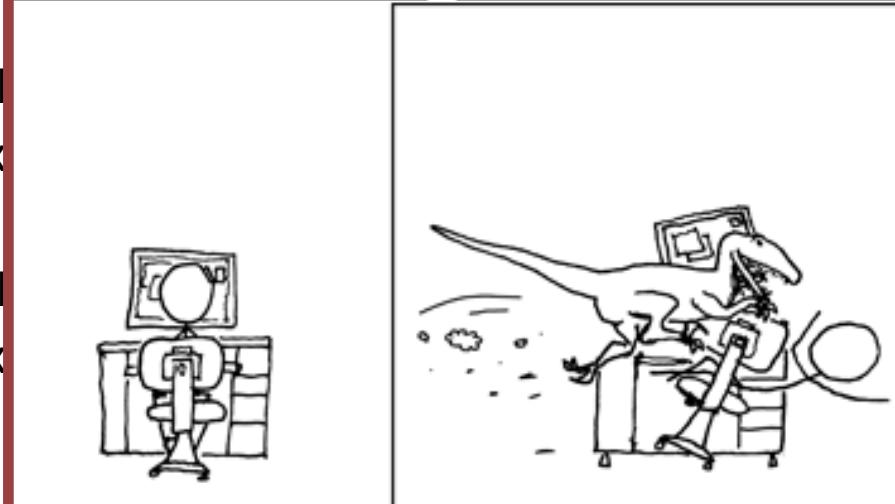
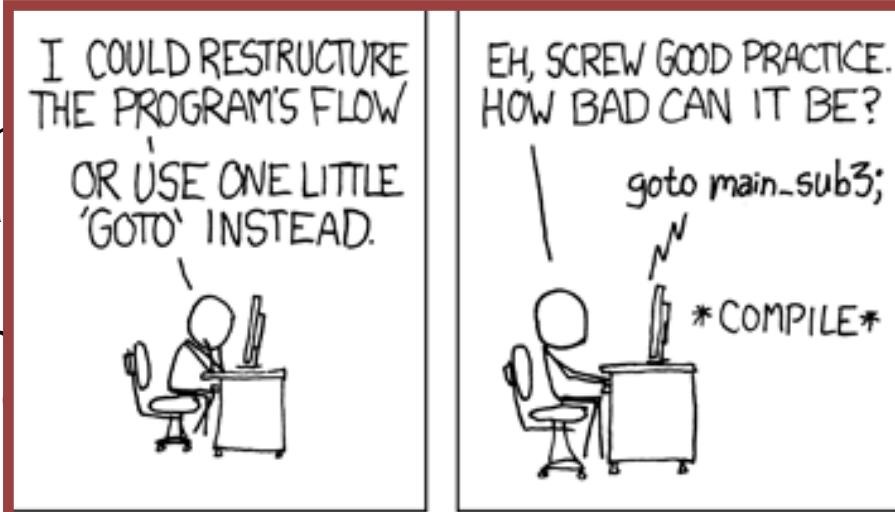
- A goto. Be careful. Can be used in two situations:
- JUMP to myself
- JUMP to an Arm
- No coordinate transformations are applied... (Meaning that if the Arms you JUMP between do not coincide you will “move” / “reorient” the neutrons...)
- Syntaxes:
- COMPONENT a=b(...)
- WHEN (expr) AT (...) JUMP somewhere
- COMPONENT a=b(...)
- WHEN (expr) AT (...) JUMP myself

***BEWARE - This IS a***



# JUMP

- A goto. Be
- JUMP to m
- JUMP to a
- No coordin
- between d
- Syntaxes:
- COMPONI
- WHEN (ex
- COMPONI
- WHEN (ex



**BEWARE - This IS a**

that if the Arms you JUMP  
the neutrons...)



# COPY- inside instruments

- In instruments: (see ILL\_H25.instr)
- COMPONENT H25\_1 = Guide\_gravity(
  - w1=0.03, h1=0.2, w2=0.03, h2=0.2, l=L\_H25\_1,
  - R0=gR0, Qc=gQc, alpha=gAlpha, m=m, W=gW)
  - AT (0,0,Al\_Thickness+gGap) RELATIVE PREVIOUS
  - ROTATED (0,Rh\_H25\_1,0) RELATIVE PREVIOUS
- COMPONENT ~~COPY(H25\_1)~~<sup>mycopy</sup> = COPY(H25\_1)
  - AT (0,0,L\_H25\_1+gGap) RELATIVE PREVIOUS
  - ROTATED (0,Rh\_H25\_1,0) RELATIVE PREVIOUS
- COMPONENT COPY(H25\_1) = COPY(H25\_1)(W=2\*gW)
  - AT (0,0,L\_H25\_1+gGap) RELATIVE PREVIOUS
  - ROTATED (0,Rh\_H25\_1,0) RELATIVE PREVIOUS



# Example: Decompose multiple scattering from Single\_crystal

```
USERVARS %{
    double multiple_scatt;
%}

...
COMPONENT Crystal = Single_crystal(... order=0 ...)
AT (0,0,0) RELATIVE somewhere
EXTEND %{
    multiple_scatt=SCATTERED;
%}

...
COMPONENT PSD_single=PSD_monitor(...)
WHEN (multiple_scatt==1) AT (0,0,0) RELATIVE somewhere_else

COMPONENT PSD_multiple=PSD_monitor(...)
WHEN (multiple_scatt > 1) AT (0,0,0) RELATIVE somewhere_else
```