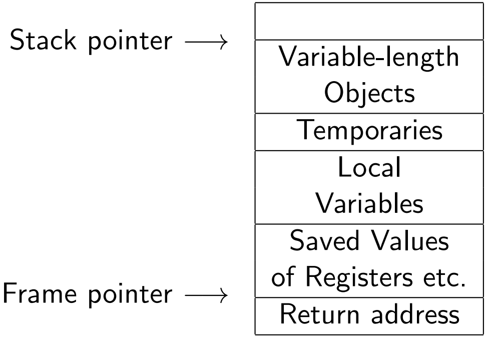
Typical Activation Record



Stack pointer ⟶

Variable-length Objects

Temporaries

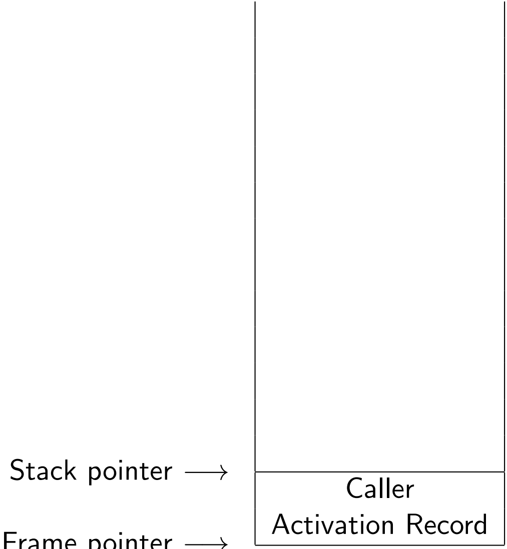
Local Variables

Saved Values of Registers etc.

Frame pointer ⟶

Return address

Calling a Subroutine



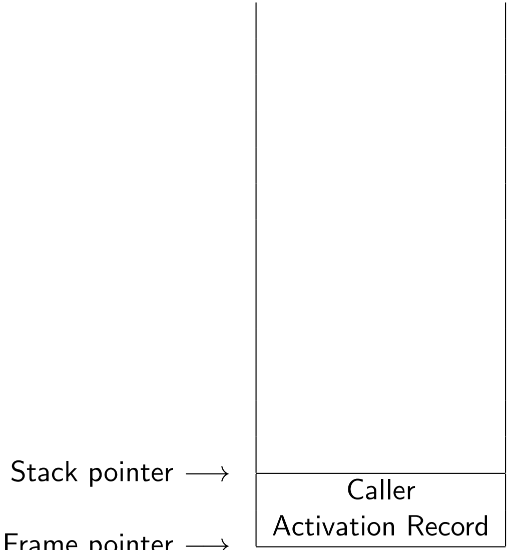
Stack pointer ⟶

Frame pointer ⟶

Caller
Activation Record

Calling a Subroutine

Calling Sequence (before)

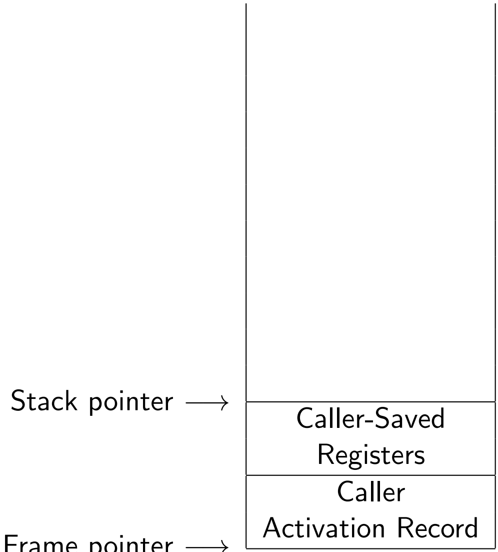


Stack pointer ⟶

Caller
Activation Record

Frame pointer ⟶

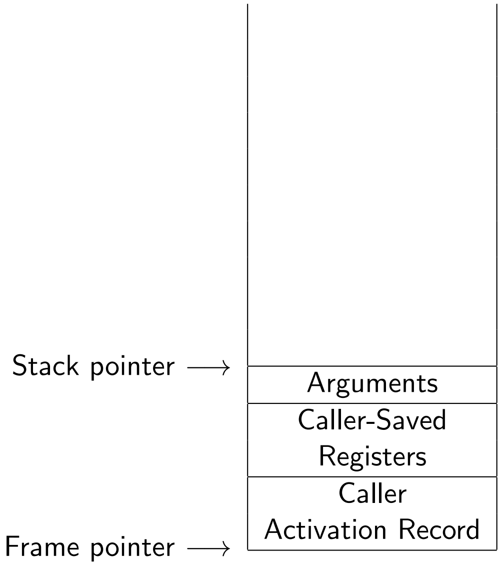
Typical Calling Sequence



Calling Sequence (before)

1. Save caller-save registers

Typical Calling Sequence


Stack pointer ⟶

Arguments

Caller-Saved Registers

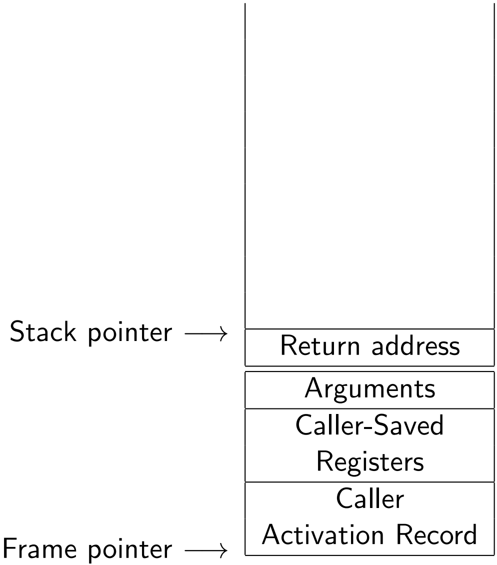
Caller Activation Record

Frame pointer ⟶

Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack

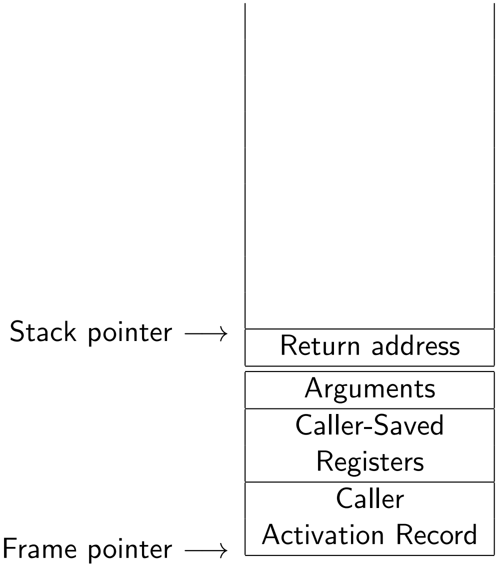
Typical Calling Sequence



Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

Typical Calling Sequence

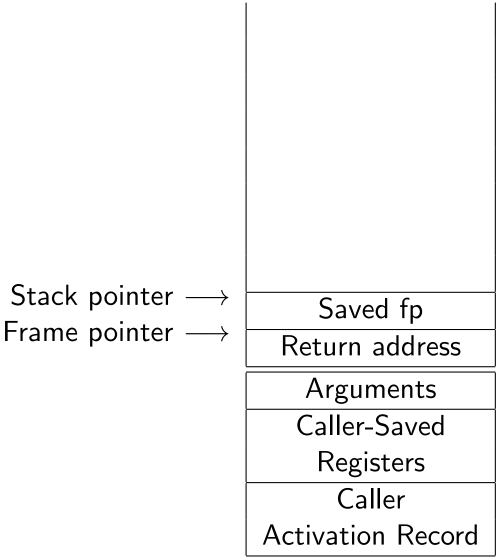


Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

Prologue

Typical Calling Sequence



Stack pointer ⟶

Frame pointer ⟶

Saved fp

Return address

Arguments

Caller-Saved Registers

Caller Activation Record

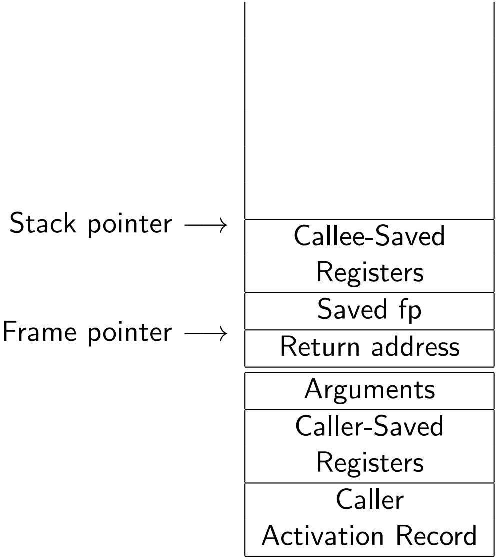
Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

Prologue

1. Save old fp, set new fp

Typical Calling Sequence



Stack pointer →

Frame pointer →

Callee-Saved Registers

Saved fp

Return address

Arguments

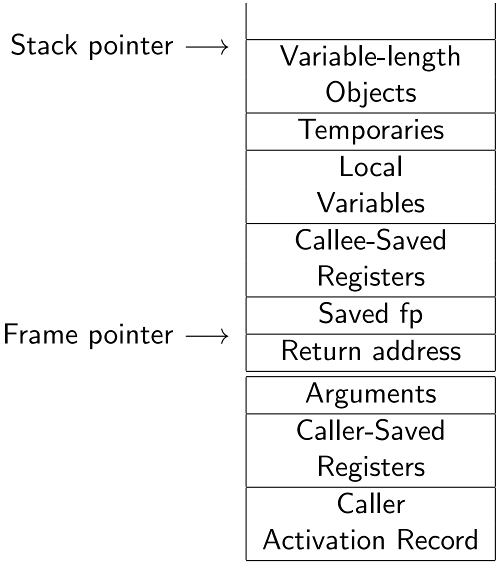
Caller-Saved Registers

Caller Activation Record

Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack

Typical Calling Sequence



Stack pointer ⟶

Variable-length Objects

Temporaries

Local Variables

Callee-Saved Registers

Saved fp

Frame pointer ⟶

Return address

Arguments

Caller-Saved Registers

Caller Activation Record

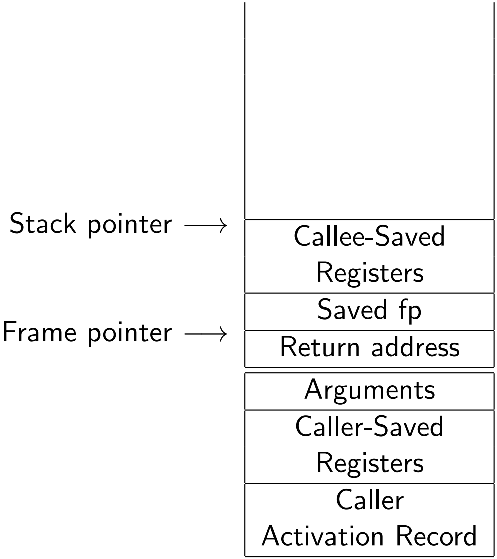
Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

Prologue

1. Save old fp, set new fp
2. Save callee-save registers

Typical Calling Sequence



Calling Sequence (before)

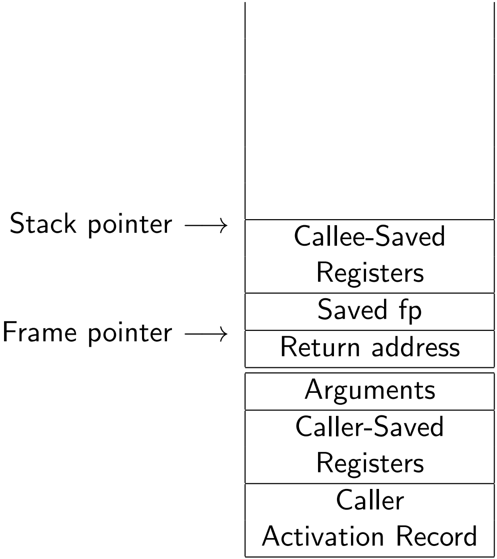
1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

Prologue

1. Save old fp, set new fp
2. Save callee-save registers

Epilogue

Typical Calling Sequence



Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

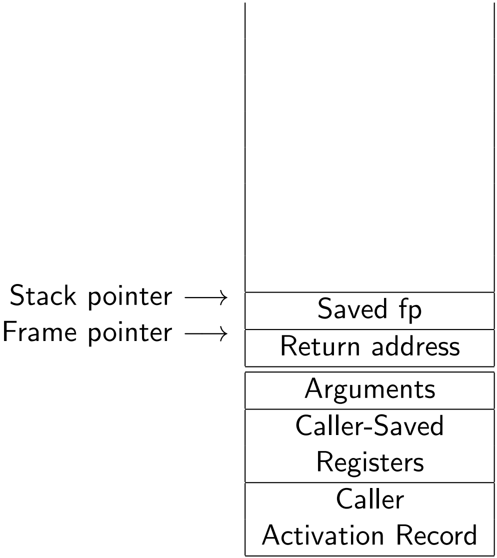
Prologue

1. Save old fp, set new fp
2. Save callee-save registers

Epilogue

1. Restore callee-save registers

Typical Calling Sequence



Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

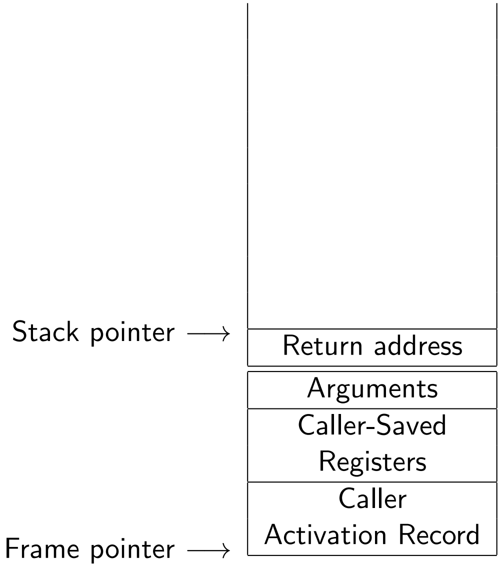
Prologue

1. Save old fp, set new fp
2. Save callee-save registers

Epilogue

1. Restore callee-save registers
2. Restore frame pointer

Typical Calling Sequence



Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

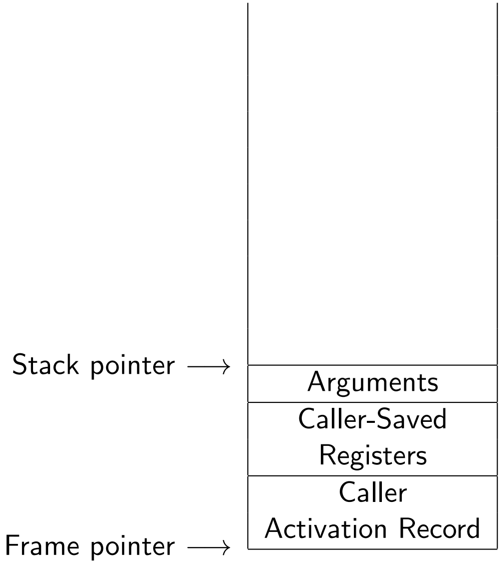
Prologue

1. Save old fp, set new fp
2. Save callee-save registers

Epilogue

1. Restore callee-save registers
2. Restore frame pointer
3. Jump to return address

Typical Calling Sequence



Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

Prologue

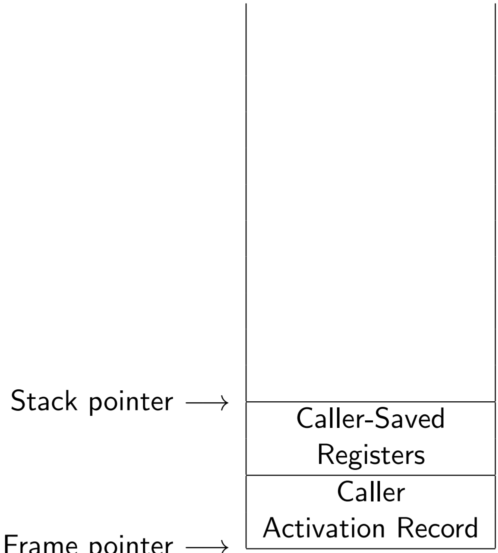
1. Save old fp, set new fp
2. Save callee-save registers

Epilogue

1. Restore callee-save registers
2. Restore frame pointer
3. Jump to return address

Calling Sequence (after)

Typical Calling Sequence



Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

Prologue

1. Save old fp, set new fp
2. Save callee-save registers

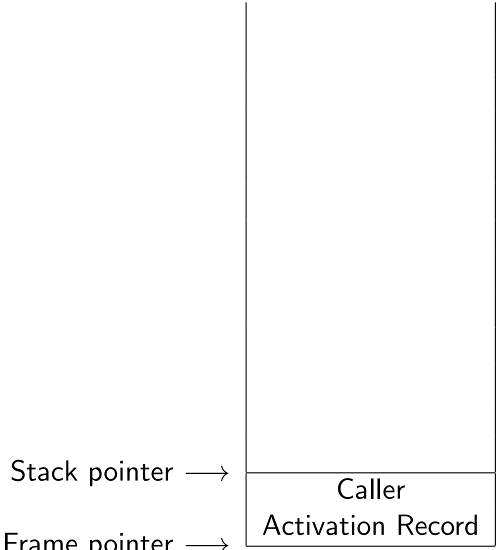
Epilogue

1. Restore callee-save registers
2. Restore frame pointer
3. Jump to return address

Calling Sequence (after)

1. Restore caller-save registers

Typical Calling Sequence



Calling Sequence (before)

1. Save caller-save registers
2. Push arguments on stack
3. Jump to subroutine, saving return address on stack

Prologue

1. Save old fp, set new fp
2. Save callee-save registers

Epilogue

1. Restore callee-save registers
2. Restore frame pointer
3. Jump to return address

Calling Sequence (after)

1. Restore caller-save registers