

System Call Index (by name)

Chapter 1: User-state System Calls

| | | |
|-----------|---|------|
| F\$Alarm | Set alarm clock..... | 1-1 |
| F\$AllBit | Send bits in an allocation map | 1-5 |
| F\$CCtl | Cache control | 1-6 |
| F\$Chain | Load and execute new primary module | 1-7 |
| F\$CmpNam | Compare two names..... | 1-9 |
| F\$CpyMem | Copy external memory..... | 1-10 |
| F\$CRC | Generate CRC | 1-11 |
| F\$DatMod | Create data module | 1-12 |
| F\$DelBit | Deallocate in bit map | 1-13 |
| F\$DExec | Execute debugged program | 1-14 |
| F\$DExit | Exit debugged program..... | 1-16 |
| F\$DFork | Fork process under control of debugger | 1-17 |
| F\$Event | Create, manipulate, and delete events..... | 1-18 |
| F\$Exit | Terminate the calling process | 1-28 |
| F\$Fork | Create a new process..... | 1-30 |
| F\$GBlkMp | Get free memory block map | 1-32 |
| F\$GModDr | Get copy of module directory | 1-34 |
| F\$GPrDBT | Get copy of process descriptor block table..... | 1-35 |
| F\$GPrDsc | Get copy of process descriptor..... | 1-36 |
| F\$Gregor | Get Gregorian date..... | 1-37 |
| F\$ID | Get process ID/user ID | 1-38 |
| F\$Icpt | Set up a signal intercept trap..... | 1-39 |
| F\$Julian | Get Julian date | 1-40 |

| | | |
|------------|--|------|
| F\$Link | Link to memory module | 1-41 |
| F\$Load | Load module(s) from a file | 1-42 |
| F\$Mem | Resize data memory area | 1-43 |
| F\$PErr | Print error message | 1-44 |
| F\$PrsNam | Parse path name | 1-45 |
| F\$RTE | Return from interrupt exception | 1-46 |
| F\$SchBit | Search bit map for a free area | 1-47 |
| F\$Send | Send a signal to another process | 1-48 |
| F\$SetCRC | Generate valid CRC in module | 1-50 |
| F\$SetSys | Set/examine OS-9 system global variables | 1-51 |
| F\$SigMask | Masks/Unmasks signals during critical code | 1-52 |
| F\$Sleep | Put calling process to sleep | 1-53 |
| F\$SPrior | Set process priority | 1-54 |
| F\$SRqCMem | System request for colored memory | 1-55 |
| F\$SRqMem | System memory request | 1-56 |
| F\$SRtMem | Return system memory | 1-57 |
| F\$SSpd | Suspend process | 1-58 |
| F\$STime | Set system date and time | 1-59 |
| F\$STrap | Set error trap handler | 1-60 |
| F\$SUser | Set user ID number | 1-62 |
| F\$SysDbg | Call system debugger | 1-63 |
| F\$Time | Get system date and time | 1-64 |
| F\$TLink | Install user trap handler module | 1-65 |
| F\$Trans | Translate memory address | 1-67 |
| F\$UAcct | User accounting | 1-68 |
| F\$UnLink | Unlink module by address | 1-69 |
| F\$UnLoad | Unlink module by name | 1-70 |
| F\$Wait | Wait for child process to terminate | 1-71 |

Chapter 2: I/O System Calls

| | | |
|-----------|---------------------------------------|------|
| I\$Attach | Attach I/O device | 2-1 |
| I\$ChgDir | Change working directory | 2-3 |
| I\$Close | Close a path to a file/device | 2-4 |
| I\$Create | Create a path to a new file | 2-5 |
| I\$Delete | Delete a file | 2-7 |
| I\$Detach | Remove a device from the system | 2-8 |
| I\$Dup | Duplicate a path | 2-9 |
| I\$GetStt | Get file/device status | 2-10 |
| I\$MakDir | Make a new directory | 2-15 |
| I\$Open | Open a path to a file or device | 2-16 |

| | | |
|-----------|---|------|
| I\$Read | Read data from a file or device | 2-18 |
| I\$ReadLn | Read a text line with editing | 2-19 |
| I\$Seek | Reposition the logical file pointer | 2-20 |
| I\$SetStt | Set file/device status | 2-21 |
| I\$Write | Write data to file or device | 2-30 |
| I\$WritLn | Write a line of text with editing | 2-31 |

Chapter 3: System-state System Calls

| | | |
|-----------|--|------|
| F\$Alarm | Set alarm clock..... | 3-1 |
| F\$AIIPD | Allocate process/path descriptor | 3-6 |
| F\$AIIPrc | Allocate process descriptor | 3-7 |
| F\$AProc | Enter process in active process queue..... | 3-8 |
| F\$DelPrc | De-allocate process descriptor service request | 3-9 |
| F\$FindPD | Find process/path descriptor | 3-10 |
| F\$IOQu | Enter I/O queue..... | 3-11 |
| F\$IRQ | Add or remove device from IRQ table | 3-12 |
| F\$Move | Move data (low bound first) | 3-14 |
| F\$NProc | Start next process | 3-15 |
| F\$Panic | System catastrophic occurrence..... | 3-16 |
| F\$RetPD | Return process/path descriptor..... | 3-17 |
| F\$SSvc | Service request table initialization | 3-18 |
| F\$VModul | Validate module | 3-20 |

System Call Index (by function)

| | | |
|-----------|--|------|
| F\$IRQ | Add or remove device from IRQ table | 3-12 |
| F\$AllBit | Send bits in an allocation bit map | 1-5 |
| F\$AllPD | Allocate process/path descriptor | 3-6 |
| F\$AllPrc | Allocate process descriptor | 3-7 |
| I\$Attach | Attach a new device to the system | 2-1 |
| F\$Cctl | Cache control | 1-6 |
| F\$SysDbg | Call system debugger | 1-63 |
| F\$Chain | Load and execute new primary module | 1-7 |
| I\$ChgDir | Change working directory | 2-3 |
| I\$Close | Close a path to a file/device | 2-4 |
| F\$CmpNam | Compare two names | 1-9 |
| F\$CpyMem | Copy external memory | 1-10 |
| F\$DatMod | Create data module | 1-12 |
| F\$Fork | Create a new process | 1-30 |
| I\$Create | Create a path to a new file | 2-5 |
| F\$Event | Create, manipulate, and delete events | 1-18 |
| F\$DelPrc | De-allocate process descriptor service request | 3-9 |
| F\$DelBit | Deallocate in bit map | 1-13 |
| I\$Delete | Delete a file | 2-7 |
| I\$Detach | Remove a device from the system | 2-8 |
| I\$Dup | Duplicate a path | 2-9 |
| F\$AProc | Enter process in active process queue | 3-8 |
| F\$I/OQu | Enter I/O queue | 3-11 |
| F\$DExec | Execute debugged program | 1-14 |

| | | |
|------------|---|------|
| F\$DExit | Exit debugged program..... | 1-16 |
| F\$FindPD | Find process/path descriptor | 3-10 |
| F\$DFork | Fork process under control of debugger | 1-17 |
| F\$CRC | Generate CRC | 1-11 |
| F\$SetCRC | Generate valid CRC in module..... | 1-50 |
| F\$GBlkMp | Get free memory block map | 1-32 |
| F\$GModDr | Get copy of module directory | 1-34 |
| F\$GPrDBT | Get copy of process descriptor block table..... | 1-35 |
| F\$GPrDsc | Get copy of process descriptor..... | 1-36 |
| F\$Gregor | Get Gregorian date..... | 1-37 |
| F\$ID | Get process ID/user ID | 1-38 |
| F\$Julian | Get Julian date | 1-40 |
| I\$GetStt | Get file/device status..... | 2-10 |
| F\$TLink | Install user trap handler module..... | 1-65 |
| F\$Link | Link to memory module | 1-41 |
| F\$Load | Load module(s) from a file | 1-42 |
| I\$MakDir | Make a new directory | 2-15 |
| F\$SigMask | Masks/Unmasks signals during critical code..... | 1-52 |
| F\$Move | Move data (low bound first) | 3-14 |
| I\$Open | Open a path to a file or device | 2-16 |
| F\$PrsNam | Parse a path name | 1-45 |
| F\$PErr | Print error message | 1-44 |
| F\$Sleep | Put calling process to sleep..... | 1-53 |
| I\$Read | Read data from a file or device..... | 2-18 |
| I\$ReadLn | Read a text line with editing | 2-19 |
| I\$Seek | Reposition the logical file pointer..... | 2-20 |
| F\$Mem | Resize data memory area | 1-43 |
| F\$RTE | Return from interrupt exception | 1-46 |
| F\$RetPD | Return process/path descriptor..... | 3-17 |
| F\$SRtMem | Return system memory | 1-57 |
| F\$SchBit | Search bit map for a free area | 1-47 |
| F\$Send | Send a signal to another process | 1-48 |
| F\$SSvc | Service request table initialization | 3-18 |
| F\$Alarm | Set alarm clock..... | 1-1 |
| F\$Alarm | Set alarm clock..... | 3-1 |
| F\$Icpt | Set up a signal intercept trap..... | 1-39 |
| F\$SPrior | Set process priority | 1-54 |
| F\$STime | Set system date and time..... | 1-59 |
| F\$STrap | Set error trap handler | 1-60 |
| F\$SUser | Set user ID number | 1-62 |

| | | |
|------------|---|------|
| F\$Time | Get system date and time | 1-64 |
| I\$SetStt | Set file/device status | 2-21 |
| F\$SetSys | Set/examine OS-9 system global variables..... | 1-51 |
| F\$NProc | Start next process | 3-15 |
| F\$SSpd | Suspend process | 1-58 |
| F\$Panic | System catastrophic occurrence..... | 3-16 |
| F\$SRqCMem | System request for colored memory | 1-55 |
| F\$SRqMem | System memory request..... | 1-56 |
| F\$Exit | Terminate the calling process | 1-28 |
| F\$Trans | Translate memory address | 1-67 |
| F\$UnLink | Unlink module by address | 1-69 |
| F\$UnLoad | Unlink module by name..... | 1-70 |
| F\$UAcct | User accounting | 1-68 |
| F\$VModul | Validate module | 3-20 |
| F\$Wait | Wait for child process to terminate..... | 1-71 |
| I\$WritLn | Write a line of text with editing | 2-31 |
| I\$Write | Write data to file or device | 2-30 |

End of System Call Index