OS Assignment 4

Joseph Bentivegna

Prof. Hakner

11/15/2017

wordgen.c Source Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <limits.h>

int main(int argc, char *argv[]) {
    int num, length;
    char randomletter;
    char word[10];
    srand(time(NULL));

    if (argc > 1) {
        num = atoi(argv[1]);
    } else {
        num = INT_MAX;
    }

    while (num > 0) {
        length = (rand() % (4)) + 3;
        for (int i = 0; i < length; i++) {
            randomletter = 'A' + (rand() % 26);
            word[i] = randomletter;
        }
        word[length] = '\0';
        printf("%s\n", word);
        num--;
    }
    return 0;
}
```

wordsearch.c Source Code:

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <ctype.h>
#include <signal.h>

int count;
```

```
void sighand(int signum) {
    fprintf(stderr, "Matched %d words\n", count);
    exit(0);
}

void makeUp(char * word) {
    for (int i = 0; i < strlen(word); i++) {
        word[i] = toupper(word[i]);
    }
}

int main(int argc, char *argv[]) {
    char *buff[500000];
    FILE *stream;
    char *word = NULL;
    size_t len = 0;
    size_t len2 = 0;
    ssize_t nread;
    ssize_t nread2;
    int ind;

    signal(SIGPIPE, sighand);

    if (argc < 2) {
        fprintf(stderr, "Not enough arguments, terminating...");
        return 1;
    } else {
        if ((stream = fopen(argv[1], "r")) == NULL) {
            fprintf(stderr, "Error opening file %s: %s\n", argv[1],
strerror(errno));
            return 1;
        }
    }

    ind = 0;
    while ((nread = getline(&buff[ind], &len, stream)) != -1) {
        makeUp(buff[ind]);
        ind++;
    }

    fclose(stream);

    count = 0;
    while ((nread2 = getline(&word, &len2, stdin)) != -1) {
        makeUp(word);
        for (int i = 0; i < ind+1; i++) {
            if (strcmp(word, buff[i]) == 0) {
                printf("%s", word);
                count++;
                break;
            }
        }
    }

    for (int i = 0; i < ind+1; i++) {
        free(buff[i]);
    }
    printf("Matched %d words\n", count);

    return 0;
}
```

pager.c Source Code:

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    FILE *term;
    int c;
    int count = 0;
    char *word = NULL;
    size_t len = 0;
    ssize_t nread;

    while ((nread = getline(&word, &len, stdin)) != -1) {
        printf("%s", word);
        count++;
        if (count == 23) {
            printf("---Press RETURN for more---");
            term = fopen("/dev/tty", "w+");
            c = getc(term);
            if (c == 113 || c == 81) exit(0);
            count = 0;
        }
    }

    return 0;
}
```

launcher.c Source Code:

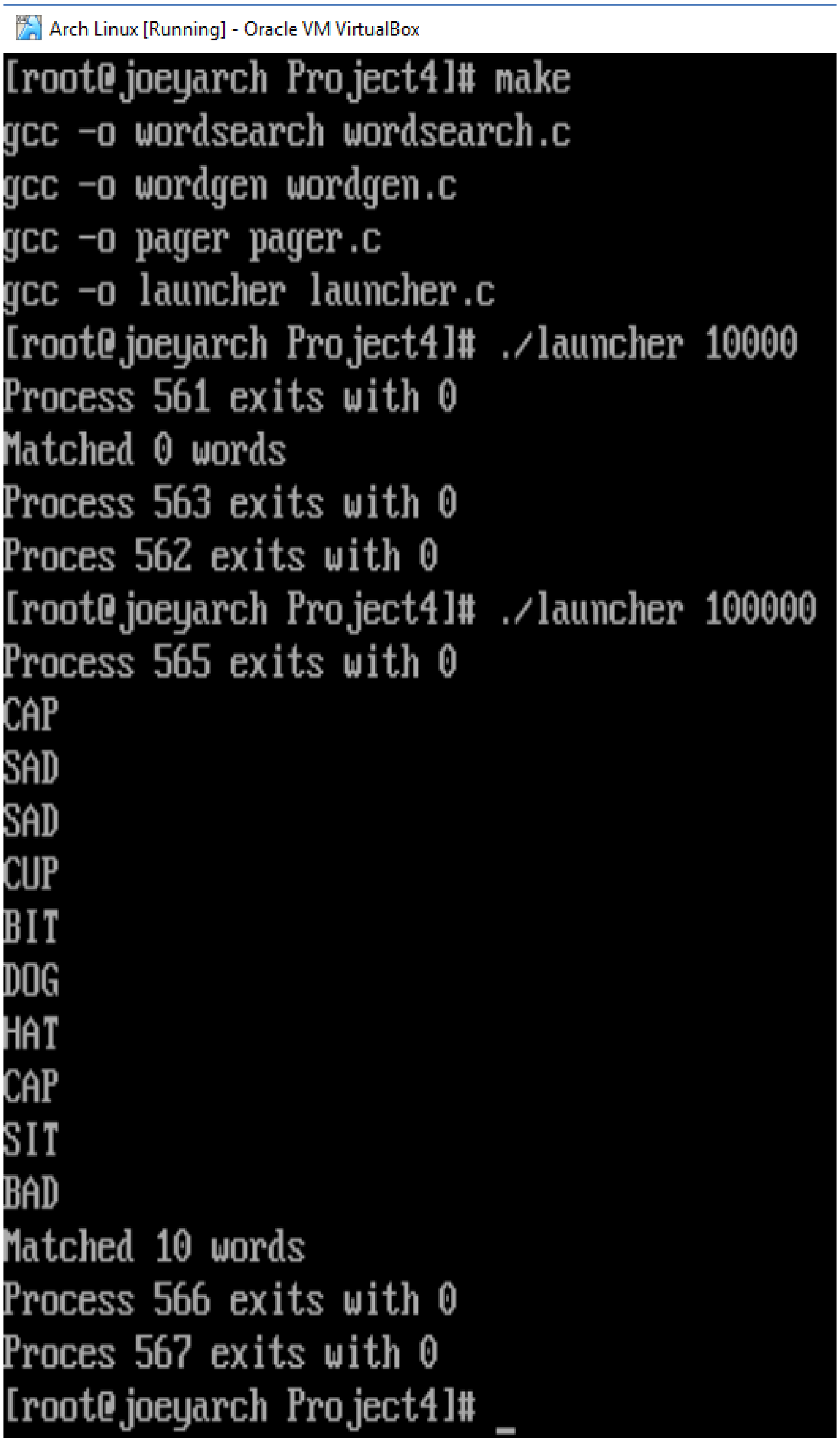
```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <errno.h>
#include <limits.h>
#include <signal.h>
#include <sys/resource.h>
#include <sys/wait.h>

int main(int argc, char *argv[]) {
    int fds[2], fds2[2];
    pid_t pid, pid2, pid3;
    pid_t opid1, opid2, opid3;
    int i, status;
    struct rusage ru;

    if (pipe(fds) < 0) {
        fprintf(stderr, "Can't create pipe (1): %s\n", strerror(errno));
        return 1;
    }

    if (pipe(fds2) < 0) {
        fprintf(stderr, "Can't creat pipe (2): %s\n", strerror(errno));
        return 1;
    }

    if ((pid = fork()) < 0) {
        fprintf(stderr, "Error forking process (1): %s\n", strerror(errno));
        return 1;
    } else if (pid == 0) {
        if (dup2(fds[1], 1) < 0) {
            fprintf(stderr, "Error duping to pipe: %s\n", strerror(errno));
            return 1;
        }
        close(fds[0]); close(fds[1]); close(fds2[0]); close(fds2[1]);
        if (execlp("./wordgen", "wordgen", argv[1], (char *)NULL) == -1) {
            fprintf(stderr, "Error executing command wordgen.c: %s\n",
strerror(errno));
        }
        exit(1);
    } else {
        if ((pid2 = fork()) < 0) {
            fprintf(stderr, "Error forking process (2): %s\n",
strerror(errno));
            return 1;
        } else if (pid2 == 0) {
            if (dup2(fds[0], 0) < 0) {
                fprintf(stderr, "Error duping to pipe: %s\n",
strerror(errno));
                return 1;
            }
            if (dup2(fds2[1], 1) < 0) {
                fprintf(stderr, "Error duping to pipe: %s\n",
strerror(errno));
                return 1;
            }
            close(fds[0]); close(fds[1]); close(fds2[0]); close(fds2[1]);
            if (execlp("./wordsearch", "wordsearch", "dict.txt", (char
*)NULL) == -1) {
                fprintf(stderr, "Error executing command wordsearch.c: %s\n",
strerror(errno));
            }
            exit(1);
        } else {
            if ((pid3 = fork()) < 0) {
                fprintf(stderr, "Error forking process (3)");
                return 1;
            } else if(pid3 == 0) {
                if (dup2(fds2[0], 0) < 0 ) {
                    fprintf(stderr, "Error duping to pipe: %s\n",
strerror(errno));
```

```
                    return 1;
                }
                close(fds[0]); close(fds[1]); close(fds2[0]); close(fds2[1]);
                if (execlp("./pager", "pager", (char *)NULL) == -1) {
                    fprintf(stderr, "Error executing command pager.c: %s\n",
strerror(errno));
                }
            }
        }
    }

    close(fds[0]); close(fds[1]); close(fds2[0]); close(fds2[1]);

    opid1 = wait3(&status, 0, &ru);
    fprintf(stderr, "Process %d exits with %d\n", opid1, status);
    opid2 = wait3(&status, 0, &ru);
    fprintf(stderr, "Process %d exits with %d\n", opid2, status);
    opid3 = wait3(&status, 0, &ru);
    fprintf(stderr, "Proces %d exits with %d\n", opid3, status);

    return 0;
}
```

<u>makefile Source Code:</u>

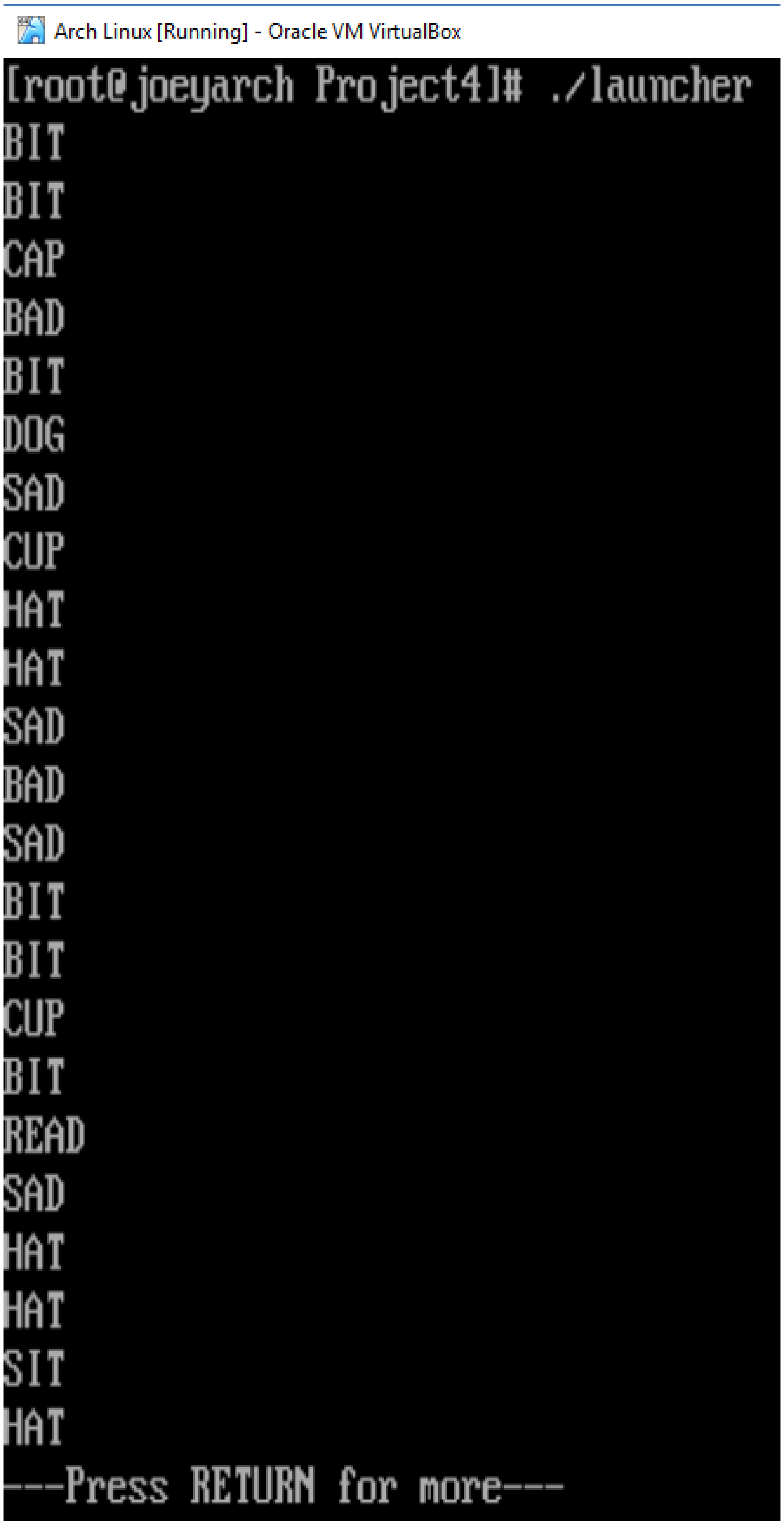
```
all: wordsearch.exe wordgen.exe pager.exe launcher.exe

wordsearch.exe: wordsearch.c
	gcc -o wordsearch wordsearch.c

wordgen.exe: wordgen.c
	gcc -o wordgen wordgen.c

pager.exe: pager.c
	gcc -o pager pager.c

launcher.exe: launcher.c
	gcc -o launcher launcher.c
```

Example Outputs:

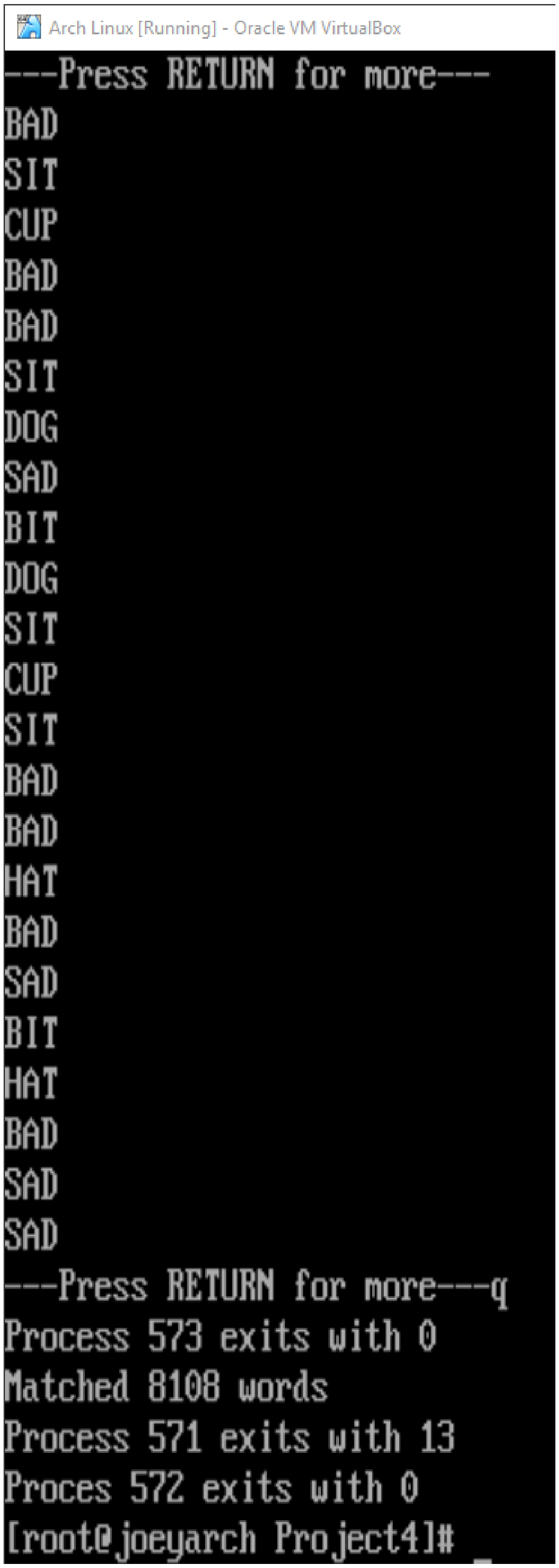


Arch Linux [Running] - Oracle VM VirtualBox

```
[root@joeyarch Project4]# make
gcc -o wordsearch wordsearch.c
gcc -o wordgen wordgen.c
gcc -o pager pager.c
gcc -o launcher launcher.c
[root@joeyarch Project4]# ./launcher 10000
Process 561 exits with 0
Matched 0 words
Process 563 exits with 0
Proces 562 exits with 0
[root@joeyarch Project4]# ./launcher 100000
Process 565 exits with 0
CAP
SAD
SAD
CUP
BIT
DOG
HAT
CAP
SIT
BAD
Matched 10 words
Process 566 exits with 0
Proces 567 exits with 0
[root@joeyarch Project4]# _
```


Arch Linux [Running] - Oracle VM VirtualBox

```
[root@joeyarch Project4]# ./launcher
BIT
BIT
CAP
BAD
BIT
DOG
SAD
CUP
HAT
HAT
SAD
BAD
SAD
BIT
BIT
CUP
BIT
READ
SAD
HAT
HAT
SIT
HAT
---Press RETURN for more---
```


Arch Linux [Running] - Oracle VM VirtualBox

```
---Press RETURN for more---
BAD
SIT
CUP
BAD
BAD
SIT
DOG
SAD
BIT
DOG
SIT
CUP
SIT
BAD
BAD
HAT
BAD
SAD
BIT
HAT
BAD
SAD
SAD
---Press RETURN for more---q
Process 573 exits with 0
Matched 8108 words
Process 571 exits with 13
Proces 572 exits with 0
[root@joeyarch Project4]# _
```